

**Project no.:**

**IST-FP6-FET-16276-2**

Project full title:

Learning to emulate perception  
action cycles in a driving school  
scenario

**Project Acronym:**

**DRIVSCO**

**Deliverable no:**

**D4.2**

**Title of the deliverable:**

**Extraction and specification of  
Structured Visual Events**

<b>Date of Delivery:</b>	5.9 2006
<b>Organization name of lead contractor for this deliverable:</b>	SDU
<b>Author(s):</b>	N Kruger, M v. Hulle, K. Pauwels, N. Chumerin, S. Kalkan, N. Pugeault, M. Lappe, F. Kandil
<b>Participant(s):</b>	SDU, KUL, UGE, UMU
<b>Work package contributing to the deliverable:</b>	WP3, WP6
<b>Nature:</b>	Brief summary with appended publications
<b>Version:</b>	1.0
<b>Total number of pages:</b>	14 pages + 12 attached documents

Project Co-funded by the European Commission		
Dissemination Level		
<b>PU</b>	Public	<b>X</b>
<b>PP</b>	Restricted to other program participants (including the Commission Services)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission Services)	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)	

## Table of Contents

1	Introduction .....	<i>p. 3</i>
2	Specification of SVEs .....	<i>p. 3</i>
3	Extraction of SVEs .....	<i>p. 5</i>
3.1	Motion and Stereo-based Events .....	<i>p. 5</i>
3.2	Independently Moving Objects (IMOs) .....	<i>p. 6</i>
3.3	SVEs Defined by Relations of Condensed Semantic Descriptors ..	<i>p. 8</i>
3.3.1	Road-based Information and Object-based Events .....	<i>p. 8</i>
3.3.2	Street Trajectory Parameterized by Polynomials .....	<i>p. 12</i>
3.3.3	Ground plane detection .....	<i>p. 13</i>



## List of Appendices

<b>Appendix A:</b> E. Baseski, N. Pugeault, S. Kalkan, D. Kraft, F. Wörgötter, and Norbert Krueger (in press). A Scene Representation Based on Multi-Modal 2D and 3D Features. Accepted for the workshop <i>3D Representation for Recognition 3dRR-07</i> .....	p. 16
<b>Appendix B:</b> Calow, D. & Lappe, M. (in press) Local statistics of retinal optic flow for self-motion through natural scenes. <i>Networks: Computations in Neural Systems</i> .....	p. 24
<b>Appendix C:</b> N. Chumerin and M. Van Hulle (in press). Cue and Sensor Fusion for Independent Moving Objects Detection and Description in Driving Scenes. <i>Signal Processing Techniques for Knowledge Extraction and Information Fusion</i> .....	p. 61
<b>Appendix D:</b> S. Kalkan, F. Wörgötter and N. Krüger (in press). First-order and second-order Statistical Analysis of 3D and 2D Structure. <i>Network: Computation in Neural Systems</i> .....	p. 83
<b>Appendix E:</b> S. Kalkan, N. Pugeault and N. Krüger (2007a). Perceptual Operations and Relations between 2D or 3D Visual Entities. Technical Report 2007 – 3 .....	p. 117
<b>Appendix F:</b> S. Kalkan, F. Wörgötter and N. Krüger (2007b). Depth Prediction at Homogeneous Image structures. Technical Report 2007 – 2 .....	p. 133
<b>Appendix G:</b> N. Krüger, N. Pugeault and F. Wörgötter. Multi-modal Primitives: Local, Condensed, and Semantically Rich Visual Descriptors and the Formalisation of Contextual Information, submitted (Technical Report 2007 – 4) .....	p. 153
<b>Appendix H:</b> K. Pauwels, M. Van Hulle (2006). Optimal instantaneous rigid motion estimation insensitive to local minima. <i>Computer Vision and Image Understanding</i> 104: 77–86 .....	p. 189
<b>Appendix I:</b> K. Pauwels and M. Van Hulle (2007a). Optic Flow from Unstable Sequences containing Unconstrained Scenes through Local Velocity Constancy Maximization. <i>BMVC</i> ...	p. 200
<b>Appendix J:</b> K. Pauwels, M. Lappe, and M. Van Hulle (2007b). Fixation as a Mechanism for Stabilization of Short Image Sequences. <i>International Journal of Computer Vision</i> 72(1), 67–78 .....	p. 211
<b>Appendix K:</b> N. Pugeault, F. Wörgötter and N. Krüger (2007). Structural Visual Events. 2007. Technical Report 2007 – 1 .....	p. 224
<b>Appendix L:</b> N. Pugeault, S. Kalkan, E. Baseski, F. Woergoetter and N. Krueger (to be submitted) Stereo Reconstruction Uncertainty and Reliability 3D-Feature .....	p. 238

## 1 Introduction

In the first 18 months we have defined and characterised the SVEs that are of relevance in the context of the project. Depending on the character of the SVE different strategies have been applied. These have been described in journal articles [Pauwels and Van Hulle (2006), Pauwels et al (2007b), Kalkan et al (accepted), Calow et al (accepted)], conferences contribution [Baseski et al 2007, Pauwels et al (2007a)], book chapters [Chumerin and M. Van Hulle (2007)] as well as in submitted work [Krueger et al. (submitted), Pugeault et al. (to be submitted)] or technical reports [Kalkan et al. 2007a, Kalkan et al. 2007b, Pugeault et al. 2007]. These works are given as appendices of this deliverable (Appendix A- Appendix L).

The following text gives an outline of our work on SVEs. For details we refer to the publications mentioned above. In section 2, we specify the SVEs we are interested in while in section 3, we give details about the algorithms we use for their extraction.

## 2 Specifications of SVEs

SVEs are defined on different levels of abstraction that then become related to SAEs on comparable levels (see Fig. 1). For example, low level SVEs such as the flow rate can be directly matched to continuous low-level SAEs such as velocity and steering angle control. SVEs of higher level of abstraction are for example crossings that require a more elaborated and abstract action and decision making process such as stopping, looking to the left and right, deciding about the risk to cross the street, etc..

We distinguish between Automatic Driving Conditions, in which vision controls action by means of closed-loop circuits, and Intentional Driving Situations, in which decisions have to be taken. Examples for Automatic Driving Conditions are straight driving, tail gaiting, curve taking, and stopping; whereas Intentional Driving Situations encompass corner taking, overtaking, obstacle avoidance (partly an automatic task!), and lane changing.

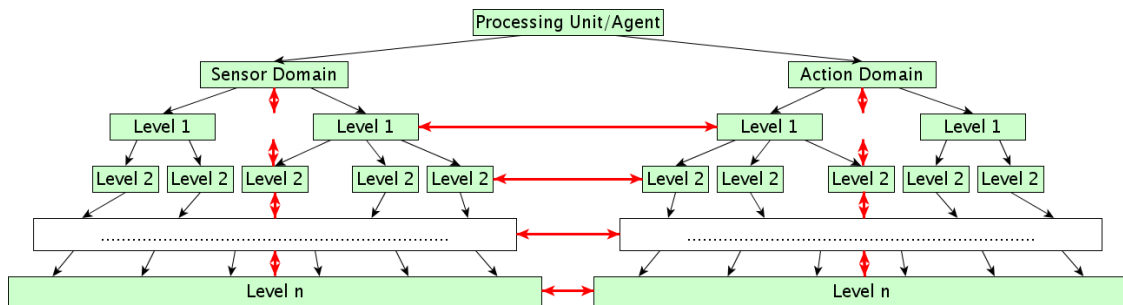


Fig. 1: Abstract view of an agent. The horizontal arrows (in red) indicate where linkage between sensor and action domain can take place. The vertical arrows indicate that sensor and action events must be seen in a certain context, which can be given by higher, or lower level

information.

Table 1 lists the different Structured Visual Events along with associated Action Events and Learning Circuits.

*Table 1: Specification of SVEs*

Structured Visual Events		SAE
<b>1</b>	<b>Motion and Stereo-based Events</b>	
1a	Flow rate	Straight driving
1b	Time to contact (relative speed or looming)	tailgating, Stopping
1c	Heading	Curve taking
1d	Curved flow lines	Curve taking
1e	Distance to objects	Straight driving, tailgating, Stopping
<b>2</b>	<b>Independently Moving Objects</b>	
2a	Number of IMOs	tailgating, Stopping
2b	Direction and speed of IMO	tailgating, Stopping
2c	Time to contact / Distance to objects	tailgating, Stopping
2d	Identity (car, truck, (motor)cycle, pedestrian)	Tailgating, Stopping
<b>3</b>	<b>Road-based information &amp; Spatial layout of the street</b>	
3a	Curvature and as derivatives: distance from beginning of the curve and tangent point	Curve Taking
3b	Lane outline and width	Straight driving, Curve taking
3c	Distance to lane (road) edges	Straight driving, Curve taking
3d	Intersections	Stopping
3e	Physical narrowing	Slowing down, precise navigation
<b>4</b>	<b>Objects</b>	
4a	Tail lights	tailgating, Stopping
4b	Traffic signs	Straight driving, Stopping
4b	Traffic lights	Straight driving, Stopping

### 3 Extraction of SVEs

According to the different SVEs described in table 1, we use different strategies for the extraction of SVEs that are briefly described in this section. In section 3.1, the extraction of motion and stereo events is outlined, in particular the ego-motion of the car. This work has been published in [Calow et al. (in press), Appendix B; Pauwels and Van Hulle (2006), Appendix H]. The extraction of independently moving objects (IMOs) is described in section 3.2. Related publications are [Pauwels and Van Hulle (2007a), Appendix I; Pauwels et al. (2007b), Appendix J]. In section 3.3, we describe the extraction of road based events and objects. Work addressing this issue has been/will be published in [Kalkan et al. (in press), Appendix D; Baseski et al. (in press), Appendix A; Chumerin and M. Van Hulle (in press), Appendix C; Krueger et al. (submitted), Appendix G; Pugeault et al. (submitted), Appendix K; Pugeault et al. (submitted), Appendix L; Kalkan et al. (2007b), Appendix E].

#### 3.1 Motion and Stereo-based Events

Motion and stereo algorithms (see Sabatini et al 2007 for the algorithms used within DrivSCO) provided present optic flow and depth maps from which flow rate, curved flow lines and depth of objects are derived. Time-to-contact can either be derived directly from optic flow maps or indirectly from object-based vision within the framework of IMO detection (see below).

Concerning heading, we have provided significant advances in the respective frameworks of the DRIVSCO proposal. With regard to the proposed analysis of the statistics of optic flow fields, we have conducted a new study dedicated to the measurement and the analysis of the statistics of optic flow generated on the retina during ego-motion through natural environments. We investigated the dependencies of the local statistics of optic flow on the environmental depth-structure, the ego-motion parameters and the position in the field of view. In order to measure these dependencies we estimated the mutual information between correlated data sets based on kernel based density estimation methods (Calow and Lappe, accepted, Appendix B). Finally, we investigated a possible link between the statistics of optical flow and receptive field properties of motion processing neurons of the Middle Temporal Area (area MT) of the primate brain.

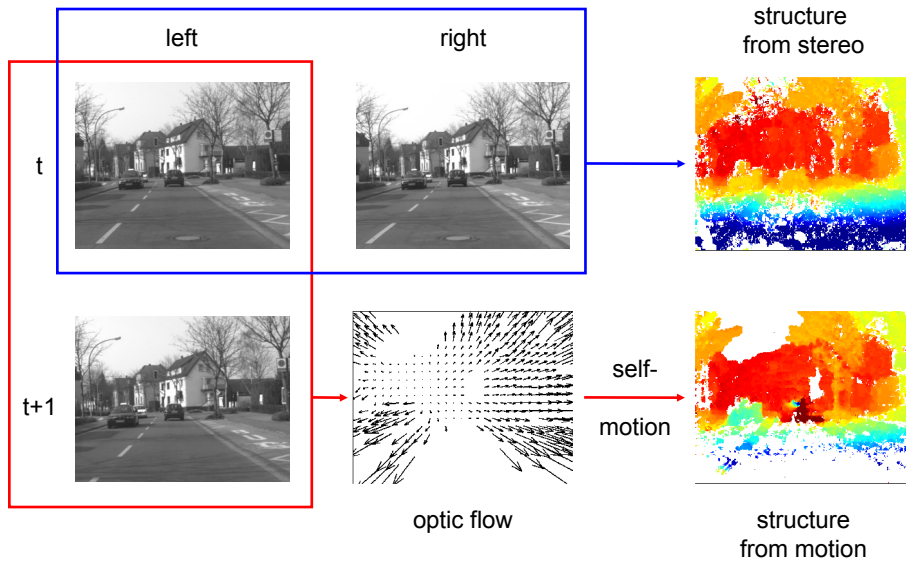
Furthermore, a new algorithm has been developed for the computation of egomotion (Pauwels and Van Hulle (2006), Appendix H). Egomotion is a crucial mid-level visual entity, as it constitutes the base for time-to-contact estimation, heading estimation, and

independent motion segmentation. This algorithm is particularly robust to local minima, while at the same time retaining the accuracy of optimal algorithms. Local minima are an important nuisance factor in the presence of independently moving objects.

Video from car-mounted cameras is particularly sensitive to jitter. Our optic flow algorithm relies on temporal consistency, which is disturbed by such instability. To compensate for this, we have developed a novel stabilization technique that is integrated within the optic flow algorithm (Pauwels and Van Hulle (2007b), Appendix J). Contrary to existing techniques, it does not rely on simplifying assumptions regarding the scene layout or type of camera motion. Our technique greatly improves the quality of the obtained optic flow.

### 3.2 Independently Moving Objects (IMOs)

The system we have developed for independent motion detection fuses optic flow, self-motion, and stereo disparity. In theory, the combination of these cues allows for the detection of all types of independent motion: those with object heading different from observer heading, those with identical direction and sign of relative heading, and those with identical direction but opposite sign of relative heading (the most difficult case). We are developing a unified approach that can detect all three types. Our approach is based on an independence hypothesis about the scene layout provided by the cues, namely the structure estimated from motion and the structure estimated from stereo. Both measures are combined in the motion field equation [Thompson and Pong (1990)]. Based on inconsistencies therein, a measure of independent motion is obtained. Figure 2 illustrates the computation of this measure.



*Fig. 2: Overview of the independent motion detection method.*

### ***Overview of the independent motion detection method:***

Two depth maps are extracted, one from optic flow and egomotion (self-motion), and the other from stereo disparity. In Fig.2 above, distance is colour-coded from blue (close) to red (far). After robustly mapping both maps onto each other, the remaining discrepancies roughly indicate the position of the moving objects (the cars in the figure). From these discrepancies, a measure of independent motion that is invariant to self-motion and environment structure is obtained. This invariance allows for temporal integration and noise-reduction at the final detection stage. A continuous stream of high-quality optic flow is required to enable this temporal integration. Video stabilization is therefore of crucial importance, and two methods have been specifically designed in the context of IMO detection. The first method [Pauwels et al. (2007b), Appendix J] not only improves the reliability of the optic flow, but also simplifies the computation of egomotion by reducing the number of parameters, consolidating information near the fovea, and increasing the number of reliable flow vectors. The second method [Pauwels and Van Hulle (2007a), Appendix I] was designed to deal with highly complex combinations of camera motion and IMOs. It has been shown to greatly increase optic flow density and reliability, even in situations of unstable camera motion in a scene that is dominated by moving objects.

Figure 3 contains a few example scenes with the detected independently moving regions marked in yellow. The locations corresponding to the moving objects are clearly marked on all occasions.

Once the independently moving regions have been identified, they can be tracked in time and disambiguated. This allows for a finer description of certain attributes of the moving regions. Examples are distance, speed, identity (car, person, bike). This is the subject of deliverable 6.2 and has been also described in [Chumerin and Van Hulle (2007)].



*Fig. 3: Independent motion detection results.*

### 3.3 Road-based Information and Object-based Events

In Ecovision (Ecovision, 2001-2003), we have derived a representation which extracts semantically rich information in terms of local multi-modal *primitives* (Krüger et al., 2004, Krueger et al (submitted), Appendix G). Multi-modal primitives provide generic information that can be applied to different kinds of problems in the context of scene analysis. However, it requires a significant amount of processing power, which will be done largely on FPGAs (WP1-WP3). Here, we will show that we can tackle different problems such as lane detection, traffic sign localization and obstacle avoidance within this representation

Road based information can also be represented by methods where vision procedures are very much designed towards a specific application. This can be very efficient and is used to detect the curvature of the lane markers and to extract the ground plane. In this context, we have developed a very fast lane marker detection system that approximates the lane as polynomials and by this gives curvature information as well as a ground plane detection system. These are described in section 3.3.2.

#### 3.3.1 SVEs Defined by Relations of Condensed Semantic Descriptors

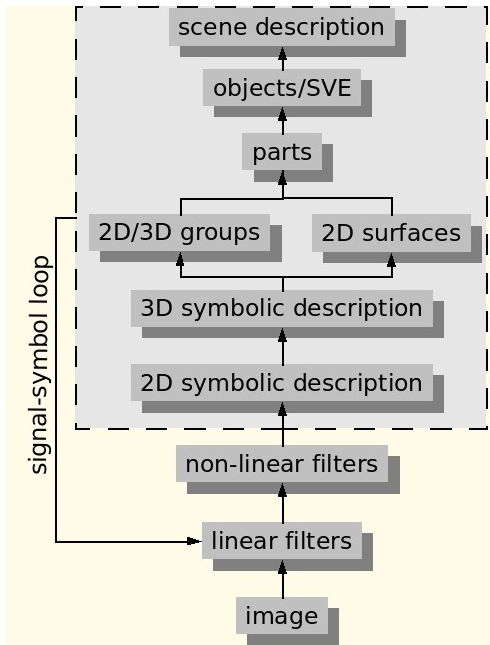


Fig 4: Proposed symbolic hierarchy.

Based on the symbolic multi-modal descriptors (Krueger et al. (submitted), Appendix G) developed mainly in the course of the ECOVISION project, we investigated an object representation framework making use of relations between collinear groups of such primitives. This approach shows some distinguishing properties:

- Information is processed over a hierarchy starting from low-level filter processes and ending with nearly textual description of objects (see figure 4). The hierarchy levels are: the original image information, a linear filtering and a non-linear filtering stage, 2D symbolic description, 3D symbolic description, 3D groups, 2D-3D

contours, parts, objects, and finally scene descriptions. The higher levels can be related to textual descriptions, allowing for semantic reasoning.

- In this hierarchy, bottom-up and top-down processes take place (see work on the signal-symbol loop in WP 3).
- This representation is rich because it covers 2D and 3D geometric as well as appearance based information. Different sources of information can be used according to its reliability and adequacy for the task at hand.
- Scene structures are represented not only as sets of local features, but also in terms of relations between entities at different levels of the hierarchy - higher level relations allowing for a making semantics explicit close to a textual description of such structures.

The Structured Visual Events (SVE) discussed herein can then be described in terms of such high level relations on their geometry and aspect. This implies the assumptions that the SVEs structures can be suitably represented as sets of primitives. Examples of valid SVEs are lane markers, traffic signs, obstacles, etc. The SVEs are defined in three steps: first, the primitives are extracted (Krüger et al. (submitted), Appendix G); second, locally collinear primitives become grouped and finally connected (see Fig. 6b and c). Additional relations between 2D and 3D primitives are defined to express relevant structural properties: co-planarity, co-colority, and parallelism – they are detailed in (Kalkan et al., 2007, Appendix E), see also figure 5. A first attempt for this has been described in (Pugeault et al., 2007, Appendix K) which becomes currently refined by a more global parameterization of semi-global 2D and 3D entities. Such descriptions are a form of “Gestalts” that can then be combined, matched against databases of known objects, and interpreted in the driving context. As shown in Fig. 6d, we have already achieved first results in scene interpretation using these Gestalts.

The semantic richness of the early cognitive vision system allows for a high level description of objects in terms of Gestalts properties, and of their relations in terms of a rather small set of properties expressed in a language-like way (two examples are given in Fig. 6d). Many objects in a traffic environment have well defined properties, e.g., street markings have a set width and distance to each other and traffic signs have a defined colour and shape. For example, in Fig. 6d the road is defined by two lanes (corresponding to Gestalts that were extracted in a bottom-up procedure) which are co-planar, parallel, and have a certain distance while the lane itself are defined by other attributes such as co-colority and co-planarity.

This prior knowledge of relevant scene structures properties can be used to search for them effectively . Since the bottom-up process that generates the Gestalts divides the scene in a relatively low order set of high level entities, an identification of objects by model knowledge can be performed rather fast. Note that this will also allow to draw correlations between SVEs and SAEs at different levels of abstraction (see Fig. 1), e.g., the lane level (e.g., lane following) and the road structure level (which can involve higher level decision processes). We have obtained first results (see Baseski et al. (in press), Appendix A) supporting further investigation of this approach scheduled for the



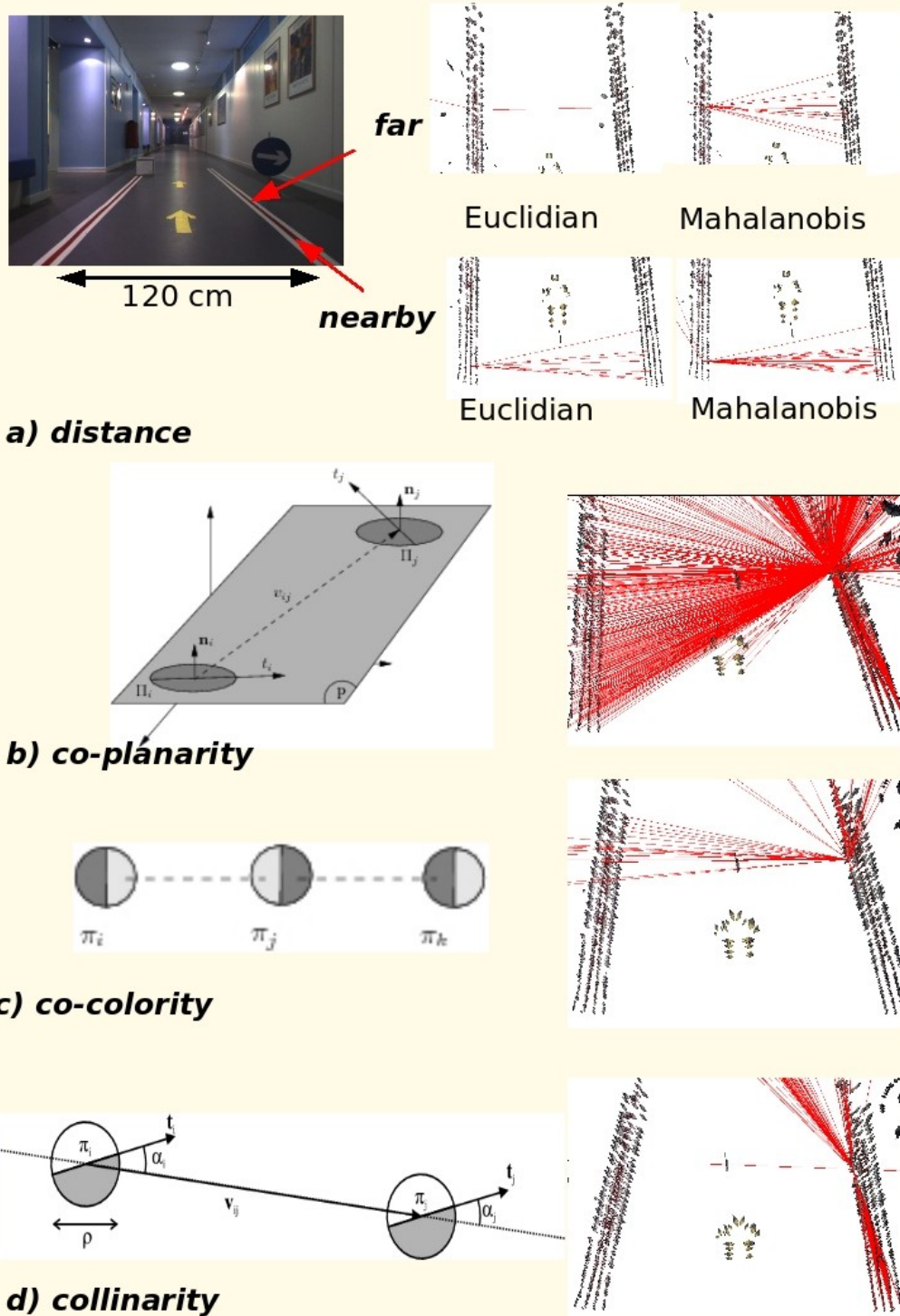


Fig. 5: Example of the relations that can be drawn between primitives. a) the distance between two primitives, using Euclidian and Mahalanobis distances. Each image shows, for one reconstructed primitive, all primitives that are at a distance of 120cm, indicated by the red lines. This is shown for nearby and far primitives, and using an Euclidian or Mahalanobis criterion. For nearby primitives both criteria pairs successfully primitives on both sides of the road, whereas for primitives farther away the Euclidian criterion pairs significantly less primitives. In b) the red lines indicate co-planar primitives (this illustrate the relative weakness of the coplanarity constraint when used on its own). In c) red lines show co-color primitives; and d) collinear primitives. In all case, red lines show all primitives paired with one chosen primitive.

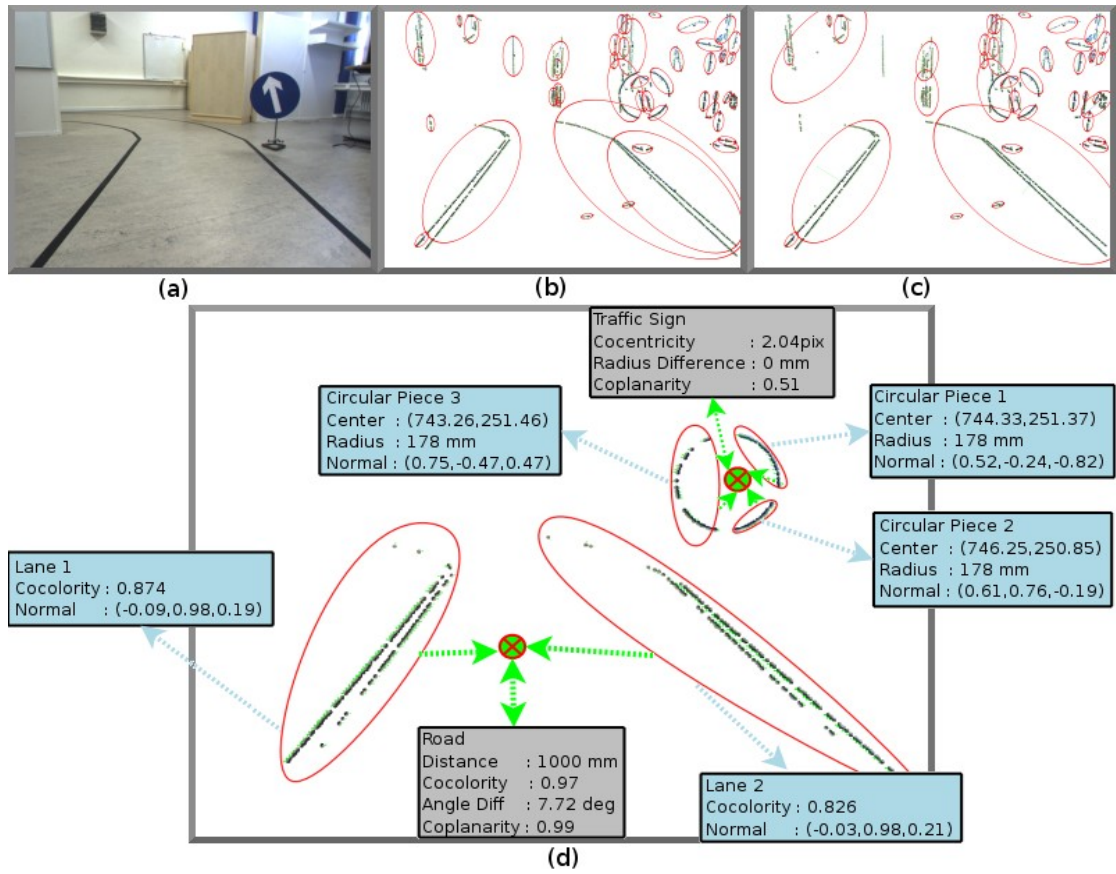


Fig. 6: Example of SVE extraction using parallelism: (a) the original image; (b) the extracted contours; (c) Gestalts being extracted partly already representing SVEs; (d) Two objects, circular traffic sign and street markers, as low order semantic combinations of Gestalts as SVEs.

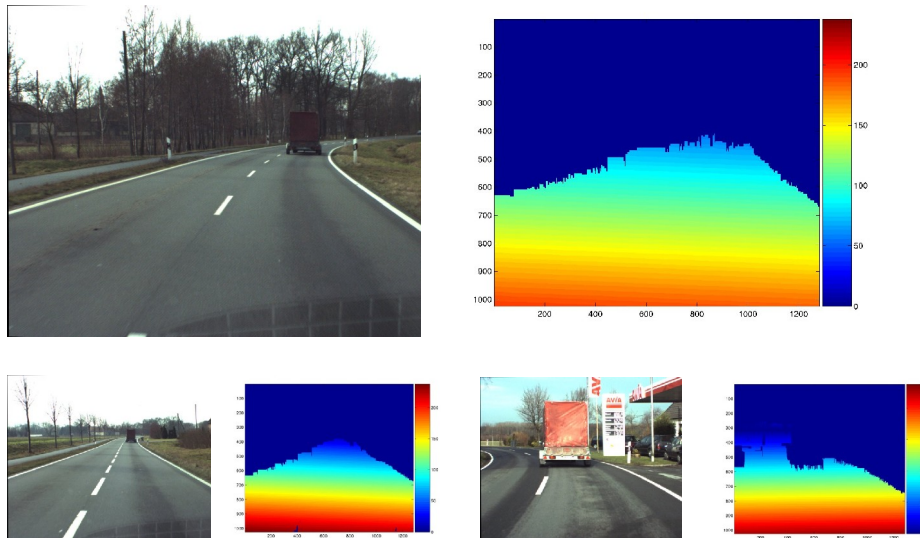


Figure 7: Three road scenes and the depth information at the road surface (shown as a disparity map), which is estimated from the depth information available at the lanes and the edges of the road.

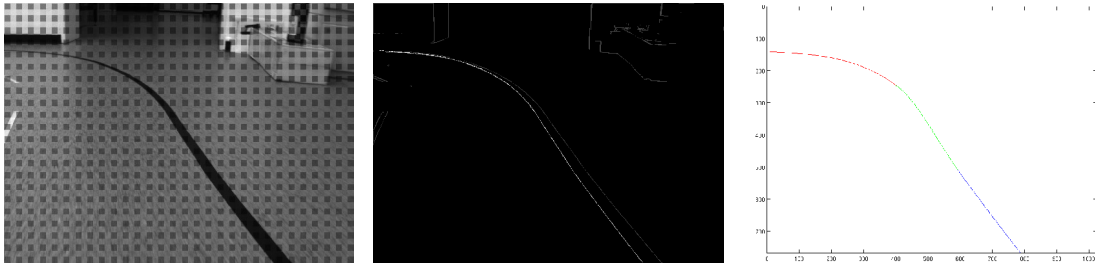
next period. Note that, since we make also use of 3D information<sup>1</sup>, we have to deal with the uncertainty thereof (see top of figure 5a where the difference between a naïve Euclidian distance measure and the use of the Mahalanobis distance is demonstrated).

Note that an explicit knowledge of the uncertainties involved in the reconstruction process is necessary to compute the Mahalanobis distance. This issue has been treated in a separate work (Pugeault et al. (to be submitted), Appendix K).

Within the representation described above, we have also addresses the issue of road estimation. It has been shown in (Kalkan et al (in press), Appendix D) using colored range images that depth at homogeneous image areas is related to the depth of the edge segments in the neighborhood. This fact is utilized in (Kalkan et al. (2007b), Appendix F) in the form of a voting-based depth prediction model, which estimates depth at homogeneous image areas from the depth of edge structures in the scene. The depth of edge structures is computed using the multi-modal primitives allows for the estimation of depth at in particular homogeneous image areas. In Fig. 7 the road structure is computed as the dominant co-planar surface for three outdoor scenes.

### 3.3.2 Street Trajectory Parameterized by Polynomials

The probably most fundamental SVE is the trajectory of the street. In order to apply any learning algorithm, street markings must be extracted and made available in an appropriate representation in real time.



*Fig. 8: Left: By the robot recorded street scenery in the lab. Middle: The image after applied edge detection, grouping and joining. The line detected as inner right street marking is plotted in white, other edges in light grey. Right: The polynomials from the output file plotted. The colours correspond to the colour coding used in the output example. Thus, the first part is red, the second green and the last blue.*

<sup>1</sup> Note that we are not dogmatic in terms of advocating the use of 3D information instead of 2D information but that we acknowledge that depending on the context and task both types of information are useful and should be accessible with the underlying uncertainties.

To detect street markings in the images recorded by the robot, e.g. as shown in Fig. 8, several steps are necessary. It must be assured that all pixels found are part of the marking, as others would add undesired noise to the description. Furthermore it is desirable to find *all* pixels that contribute to the street marking, as any less would be a source of noise. For this we use the common assumption that the longest line in the image describes the street lane marking. Unfortunately noise in the recorded images, reflections, and occlusions provoke edges interruptions. Thus, the longest line in the edge image, indicating the street marking, may decay into short line segments. The methods used to deal with these interruptions are very technical and shall not be described here. The result is a nearly un-fragmented right lane marking, where detection of the left lane is left for further processing. Note, the left lane is often not visible in the image, or only a short part of it, due to the limited field of view allowed by the deployed lenses.

The detected lane is then represented by a function fitting method where we approximate the line piecewise with three polynomials. The algorithm outputs are the polynomials parameters, a timestamp and an indication whether the detected marking describes the left or the right lane. Three polynomials are used because every single one tends to diverge at the outer limits of the curve. An interval is given which indicates where the given polynomial is defined. In the example above, polynomials for different coefficients are drawn, and colour coded as follows:

part 0:  $139.835 + 0.0660085*x - 0.000173041*x.^2 + 1.64977e-06*x.^3$   
part 1:  $2400.2 - 14.5005*x\_2 + 0.0304645*x\_2.^2 - 1.92065e-05*x\_2.^3$   
part 2:  $-84.8001 + 0.32184*x\_3 + 0.00171013*x\_3.^2 - 9.44763e-07*x\_3.^3$

### 3.3.3 Ground plane detection

In order to estimate the ground plane, we estimate the disparity plane, then map the set of points from the disparity domain into a 3D world domain, and finally fit a plane through the projected set (a detailed description is given in [Chumerin and Van Hulle (in press), Appendix C]).

Before the disparity plane estimation, we intersect the disparity map with the pre-defined road mask (see Fig. 9, left panel). By this step, we filter out the majority of pixels which do not belong to the ground plane and are outliers in the disparity plane linear model.

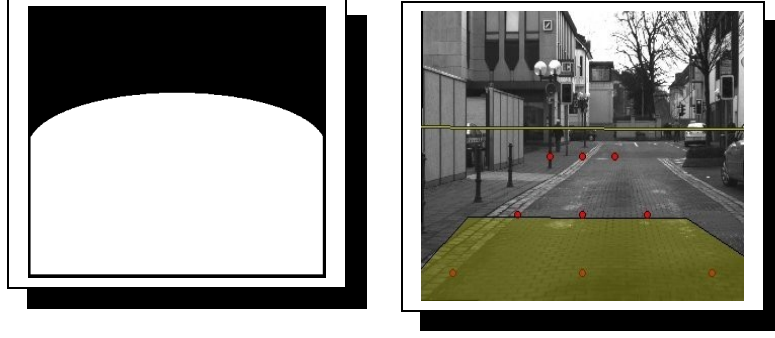


Figure 9: Ground plane detection

The disparity plane parameters are estimated using IRLS (Iteratively Reweighted Least-Squares with weight function proposed by (Beaton 1974)). For the ground plane parameters estimation, we choose a set of nine points 3x3 lattice) in the lower half of the frame (see Fig. 9, right panel). Disparities for these points are determined using the estimated disparity plane. Given the disparities and camera calibration data, we project the selected points into a 3D world coordinate system. In addition, we add two so-called *stabilization points* which correspond to the points where the front wheels of the test car are supposed to touch the road surface. For the inverse projection of the stabilization points, we use parameters of the canonic *disparity plane*: it is a disparity plane which corresponds to the horizontal ground plane observed by cameras in a quiescent state. The parameters of the canonic disparity plane and positions of the stabilization points were obtained based on the test car geometry and camera setup position and orientation in the test car.

The full set of 11 points is then used for IRLS fitting of the ground plane in a world coordinate system. During the disparity plane estimation, we use the estimation from the previous frame for weight initialization in IRLS; for the first frame, for the same purpose, we use the parameters of the canonic disparity plane. We assume that the ground plane is estimated correctly if its orientation has a quite small deviation (norm of difference of the unity normal vectors) from the orientation of the canonic ground plane and in the same time from the orientation of the plane obtained at the previous frame. Otherwise the estimation from the previous frame is used.

## References

ECOVISION (2003-2005). Artificial Visual Systems based on early-cognitive cortical Processing, EU-Project.

A.E. Beaton and J.W. Tukey (1974). The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. *Technometrics*, 16 (2):147–185.

Chumerin, N. and Van Hulle, M.M. (in press). An Approach to On-Road Vehicle Detection, Description and Tracking. *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*.

N. Kruger, M. Lappe and F. Worgotter (2004). Biologically Motivated Multi-modal Processing of Visual Primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour* 1(5): 417-428.

S. P. Sabatini, G. Gastaldi, F. Solari, K. Pauwels, M. v. Hulle, J. Diaz, E. Ros, N. Pugeault and N. Krüger (2007). Compact (and Accurate) Early Vision Processing In The Harmonic Space. *Int. Conf. on Computer Vision Theory and Applications. VISAPP*.

Thompson, W.B. and Pong, T.C. (1990). Detecting Moving Objects. *International Journal of Computer Vision* 4 (1): 39-57.

## **Appendix A**

### ***A Scene Representation Based on Multi-Modal 2D and 3D Features***

E. Baseski, N. Pugeault, S. Kalkan, D. Kraft, F. Wörgötter, and Norbert Krueger  
( *Accepted for the workshop 3D Representation for Recognition 3dRR-07* )



# A Scene Representation Based on Multi-Modal 2D and 3D Features

Emre Başeski  
Syddansk Universitet  
Denmark

emre@mmmi.sdu.dk

Dirk Kraft  
Syddansk Universitet  
Denmark

kraft@mmmi.sdu.dk

Nicolas Pugeault  
University of Edinburgh  
United Kingdom

npugeaul@inf.ed.ac.uk

Florentin Wörgötter  
Universität Göttingen  
Germany

worgott@bccn-goettingen.de

Sinan Kalkan  
Universität Göttingen  
Germany

sinan@bccn-goettingen.de

Norbert Krüger  
Syddansk Universitet  
Denmark

norbert@mmmi.sdu.dk

## Abstract

*Visually extracted 2D and 3D information have their own advantages and disadvantages that complement each other. Therefore, it is important to be able to switch between the different dimensions according to the requirements of the problem and use them together to combine the reliability of 2D information with the richness of 3D information. In this article, we use 2D and 3D information in a feature-based vision system and demonstrate their complementary properties on different applications (namely: depth prediction, scene interpretation, grasping from vision and object learning)<sup>1</sup>.*

## 1. Introduction

There exist acknowledged differences between visually extracted 2D and 3D information (see, e.g., [2, 4]). In addition to the difference in dimension, two aspects of 2D information can be distinguished [12]: appearance based information (such as pixel color values or contrast transition) and geometric information (such as the position and orientation of a local edge). An overview of such differences is given in Table 1.

Two dimensional geometric information varies significantly with viewpoint changes. Actually, it is only the change of 2D orientation that allows for the reconstruction of a 3D orientation. For many tasks such as object recognition, this imposes the problem to compensate for this variance which can be done for example by invariant descriptors (see, e.g., [10, 11]). However, an invariance to such transformations leads necessarily to a weakening of the structural richness of the representations since properties that the

system becomes invariant to can not be represented anymore.

For both types of 2D information, geometric or appearance based, the transformation under viewpoint changes can be computed explicitly or at least approximated once the underlying 3D model is known. Hence, using 3D information reduces the problem of variance under view-point transformation (with the exception of occlusions) and also allows to compute rich geometric information in terms of 3D position and 3D orientation. It also allows for the definition of semantic relations such as the Euclidian distance of visual entities or their co-planarity (see below). Moreover, in the context of robotic systems, the 3D space is closer to the space the action takes place in comparison to the 2D image space. For example in grasping, the transformation between joint co-ordinates and 3D pose is usually trivial [13]; and in navigation, planning is often done in maps representing depth information in an Euclidian way.

However, there are also problems connected to the use of 3D information. First, significantly more complex processing is required: Besides the fact that multiple cameras are required that usually need to be carefully calibrated, correspondences need to be found. For feature based matching, this imposes a number of possible error sources. For example, besides the possibility of a wrong match, it might even be that a feature is extracted in only one of the images. Moreover, when 3D information is extracted by stereo, the quality of information highly varies *with* space since the uncertainties that are associated to reconstructions at different positions in Euclidian space are highly non-isotropic and hence any depth information carries an uncertainty that depends strongly on the viewpoints [15].

We suggest that efficient visual systems should make use of the complementary properties of 2D and 3D information according to the actual context and task. This seems to hold

<sup>1</sup>This work has been supported by EU-Project Drivscio



for human vision as well. For example, although 2D information is sufficient for a large number of vision tasks, Edelman and Bülthoff [4] have shown that the existence of 3D information reduces the mean error rate for tasks like recognition. Since 2D information is more reliable but 3D information is richer, one can for example use the complementary aspects of both kinds of information by doing semantic reasoning and hypotheses generation in the 3D space and feed these hypotheses back to lower levels of processing.

In [9], a visual representation, which is based on local symbolic features called multi-modal primitives, has been introduced. These primitives (see Figure 1(a)) represent a local part of the scene in terms of condensed 2D and 3D information covering appearance based aspects of visual information (color and local phase) as well as geometric information in terms of 2D and 3D position and orientation. These primitives allow for switching between 2D and 3D as well as geometric and appearance based information and hence their complementary properties can be used efficiently. Moreover, in [15], a model for the uncertainties of the 3D properties covered by the primitives is derived and is used to facilitate the reasoning processes in 3D space.

Originally, the multi-modal primitives have been designed to formulate predictions in an early cognitive vision system to disambiguate visual information (see [19]). In this work, we make use of this representation to characterize scenes and objects by 2D and 3D properties of the primitives as well as by a number of relations defined upon the primitives such as parallelism, co-planarity etc. We show that the structural richness of the representations allows for semantic reasoning about object properties and object relations in scenes. The representations are rather generic since they basically cover known attributes of visual information such as orientation, color, local motion as also computed in the first stages of human visual processing [7].<sup>2</sup> Hence, the primitives can be made use of for a variety of tasks.

In this paper, the strength of the approach is demonstrated on a variety of applications such as depth prediction, road interpretation, grasping, and object learning. Here, we focus less on the detailed description of the algorithms but on how the introduced representation facilitates the computation for the different tasks. In that sense, this article has a review character of previous works as well.

The paper is structured as follows: In section 2, the visual representation in [9] is summarized. In section 3, we then briefly describe 4 applications and in section 4, we reflect upon the properties of the representation.

<sup>2</sup>A more detailed discussion of the biological motivation can be found in [9].

## 2. Primitives and Relations

In [9], a visual representation has been introduced in terms of local condensed symbolic features called multi-modal primitives. We give a brief description of these features in section 2.1. In section 2.2, we introduce perceptual relations on these symbolic features that are applied in the applications described in section 3.

### 2.1. Multi-modal primitives

In its current state, the primitives discussed can be edge-like or homogeneous and carry 2D or 3D information. For edge-like primitives, the corresponding 3D primitive is extracted using feature based stereo. Since correspondences can not be found for homogeneous image structures, 3D primitives for these image structures can be estimated from the surrounding 3D edge-like primitives (see also section 3.1).

An edge-like 2D primitive (Figure 1(a)) is defined as:

$$\pi = (\mathbf{m}, \theta, \omega, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r), f), \quad (1)$$

where  $\mathbf{m}$  is the image position of the primitive;  $\theta$  is the 2D orientation;  $\omega$  represents the contrast transition coded in the local phase;  $(\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$  is the representation of the color, corresponding to the left ( $\mathbf{c}_l$ ), the middle ( $\mathbf{c}_m$ ) and the right side ( $\mathbf{c}_r$ ) of the primitive; and,  $f$  is the optical flow.

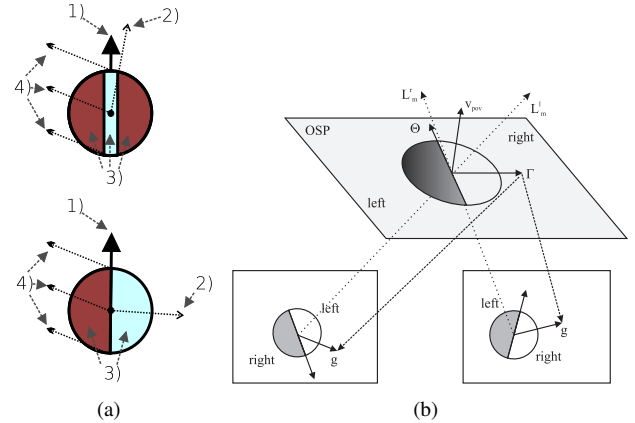


Figure 1. (a) Two types of edge-like 2D primitives [9] 1) represents the orientation of the primitive, 2) the phase, 3) the color and 4) the optic flow. (b) Reconstruction of a 3D primitive from two 2D primitives.

As the underlying structure of an homogeneous image patch is different from that of an edge-like patch, a different representation is needed for homogeneous 2D primitives (called *monos*):

$$\pi^m = (\mathbf{m}, \mathbf{c}), \quad (2)$$

where  $\mathbf{m}$  is the position in the image, and  $\mathbf{c}$  is the color of the mono. Note that these different image structures can be distinguished by the intrinsic dimension of the image patch

	3D	2D	
pros	Distances and angles are invariant under camera transformations	Distances and angles are variant under camera transformations	cons
	Units have physical meaning (distance in millimeters)	Pixel coordinates are not directly usable for physical measurements	
	Relations are richer (coplanarity, proximity)	Restricted to 2D relations	
	Possible to obtain a complete model of an object	To cover all perspectives of an object a high number of images are required	
	Directly relatable to actions	Requires additional computation to become related to actions	
cons	High computational complexity	Low computational complexity	pros
	High likelihood of errors and uncertainty	Higher reliability	

Table 1. Different properties of 2D and 3D information. While 3D information has geometric properties (position and orientation), 2D information covers also appearance based properties (color, contrast transition etc.).

[5]. See [9] for more information about these modalities and their extraction. Figure 2 shows the extracted primitives for an example scene.

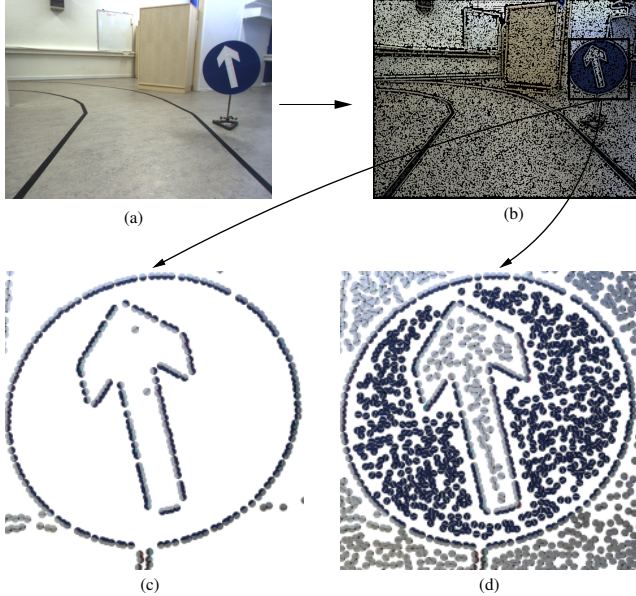


Figure 2. Extracted primitives (b) for the example image in (a). Magnified edge primitives and edge primitives together with monos are shown in (c) and (d) respectively.

A primitive  $\pi$  is a 2D feature which can be used to find correspondences in a stereo framework to create 3D primitives (as introduced in [16]) which have the following formulation:

$$\Pi = (M, \Theta, \Omega, (c_l, c_m, c_r)), \quad (3)$$

where  $M$  is the 3D position;  $\Theta$  is the 3D orientation. Appearance based information is coded in the phase  $\Omega$  (i.e., contrast transition) and  $(c_l, c_m, c_r)$  is the representation of

the color, corresponding to the left ( $c_l$ ), the middle ( $c_m$ ) and the right side ( $c_r$ ) of the 3D primitive. Both, phase and color, are extracted as a combination of the associated values in the corresponding 2D primitives in the left and right image. The reconstruction of a 3D primitive from two corresponding 2D primitives is exemplified in Figure 2(b).

In section 3.1, we estimate the 3D representation  $\Pi^m$  of monos which stereo fails to compute:

$$\Pi^m = (M, \mathbf{n}, \mathbf{c}), \quad (4)$$

where  $M$  and  $\mathbf{c}$  are as in equation 2, and  $\mathbf{n}$  is the orientation (i.e., normal) of the plane that locally represents the mono.

## 2.2. Perceptual relations between primitives

The sparse and symbolic nature of the discussed primitives allows for perceptual relations defined on them that express relevant spatial relations in 2D and 3D space. These relations can be applied in rather different contexts such as depth prediction, object learning and grasping (see section 3).

**Collinearity:** Two spatial primitives  $\Pi_i$  and  $\Pi_j$  are collinear (i.e., part of the same group) if they are part of the same contour. Due to uncertainty in the 3D reconstruction process, in this work, the collinearity of two spatial primitives  $\Pi_i$  and  $\Pi_j$  is computed using their 2D projections  $\pi_i$  and  $\pi_j$ . We define the collinearity of two 2D primitives  $\pi_i$  and  $\pi_j$  as:

$$col(\pi_i, \pi_j) = 1 - \left| \sin \left( \frac{|\alpha_i| + |\alpha_j|}{2} \right) \right|, \quad (5)$$

where  $\alpha_i$  and  $\alpha_j$  are as shown in Figure 3(a).

**Co-planarity:** Two 3D edge primitives  $\Pi_i$  and  $\Pi_j$  are defined to be co-planar if their orientation vectors lie on the same plane, i.e.:

$$cop(\Pi_i, \Pi_j) = 1 - |\mathbf{proj}_{t_j \times \mathbf{v}_{ij}}(t_i \times \mathbf{v}_{ij})|, \quad (6)$$

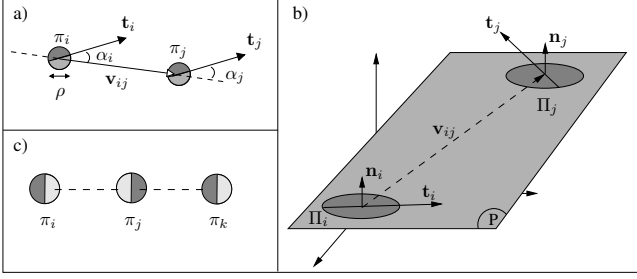


Figure 3. Illustration of the perceptual relations between primitives. (a) Collinearity of two 2D primitives. (b) Co-colority of three 2D primitives  $\pi_i$ ,  $\pi_j$  and  $\pi_k$ . In this example,  $\pi_i$  and  $\pi_j$  are cocolor, so are  $\pi_i$  and  $\pi_k$ ; however,  $\pi_j$  and  $\pi_k$  are not cocolor. (c) Co-planarity of two 3D primitives  $\Pi_i$  and  $\Pi_j$ .

where  $\mathbf{v}_{ij}$  is the vector  $(M_i - M_j)$ ;  $t_i$  and  $t_j$  denote the vectors defined by the 3D orientations  $\Theta_i$  and  $\Theta_j$ , respectively; and,  $\text{proj}_{\mathbf{u}}(\mathbf{a})$  is the projection of vector  $\mathbf{a}$  over vector  $\mathbf{u}$ . The co-planarity relation is illustrated in Figure 3(b).

**Co-colority:** Two 3D primitives  $\Pi_i$  and  $\Pi_j$  are defined to be co-color if their parts that face each other have the same color. In the same way as collinearity, co-colority of two spatial primitives  $\Pi_i$  and  $\Pi_j$  is computed using their 2D projections  $\pi_i$  and  $\pi_j$ . We define the co-colority of two 2D primitives  $\pi_i$  and  $\pi_j$  as:

$$\text{coc}(\pi_i, \pi_j) = 1 - \mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j), \quad (7)$$

where  $\mathbf{c}_i$  and  $\mathbf{c}_j$  are the RGB representation of the colors of the parts of the primitives  $\pi_i$  and  $\pi_j$  that face each other; and,  $\mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j)$  is Euclidean distance between RGB values of the colors  $\mathbf{c}_i$  and  $\mathbf{c}_j$ . Co-colority between an edge primitive  $\pi$  and a mono primitive  $\pi^m$ , and between two monos can be defined similarly (not provided here). In Figure 3(c), a pair of co-color and not co-color primitives are shown.

**Rigid-body motion:** The rigid body motion  $\mathcal{M}_{t \rightarrow t+\Delta t}$  associating any entity in space in the coordinate system of the stereo set-up at time  $t$  to the same entity in the new coordinate at time  $t + \Delta t$  is explicitly defined for 3D-primitives (see Figure 4):

$$\hat{\Pi}_i^{t+\Delta t} = \mathcal{M}_{t \rightarrow t+\Delta t}(\Pi_i^t). \quad (8)$$

### 3. Applications

In this section, the framework introduced in section 2 is applied to a variety of tasks such as depth prediction at homogeneous image structures (section 3.1), scene interpretation (section 3.2), grasping (section 3.3) and object learning (section 3.4).

#### 3.1. Depth prediction

Edge primitives represent edge-like structures. It is known that it becomes increasingly difficult to find corre-

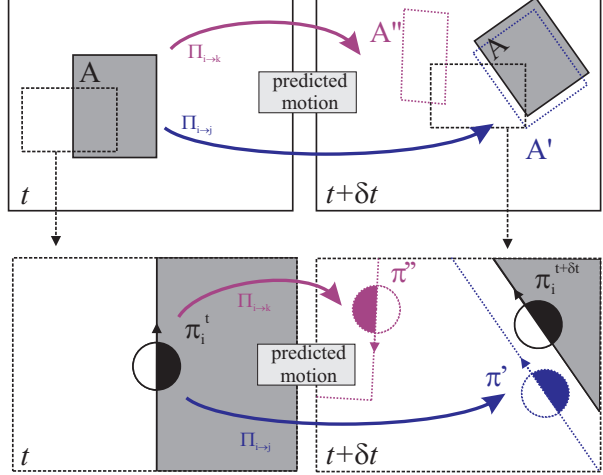


Figure 4. Example of the rigid-body motion of a primitive (see text).

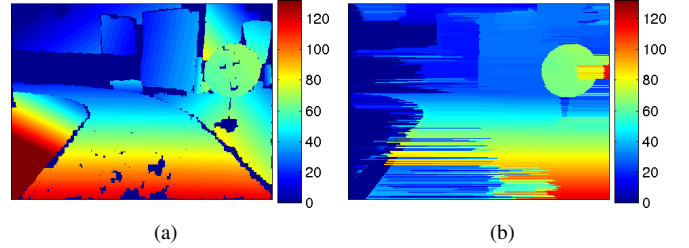


Figure 5. Depth prediction at homogeneous image areas using perceptual relations between primitives. (a) The results, shown as a disparity map only at the predictions, are from the scene in Figure 2. (b) A global dense stereo method (taken from [18]) that uses dynamic programming to optimize matching costs.

spondences between local patches the more they lack structure. On the other hand, it is known that lack of structure also indicates lack of a depth discontinuity [6, 8]. Moreover, we have shown that based on the co-planarity relation, depth at homogeneous image areas can be predicted (see Figures 5 and 6). Such a scheme can be used to ‘fill in’ the representation at homogeneous areas using co-planar relationships between edge-like primitives. In Figure 5, the homogeneous primitives inferred using such a scheme are shown as a disparity map. Results on the same scene are shown for a global dense stereo method (taken from [18]) that uses dynamic programming to optimize matching costs. Figure 5 shows that such depth prediction can be used as a depth cue providing additional information in particular when image structures are too weak to find correspondences. When confronted with an image as in Figure 6, many dense depth estimation algorithms either basically fail or assume implicitly some linearity assumption that leads to rather bad reconstruction. However, our method can ‘interpret’ the curved edges of the cylinder in order to reconstruct the round surface.

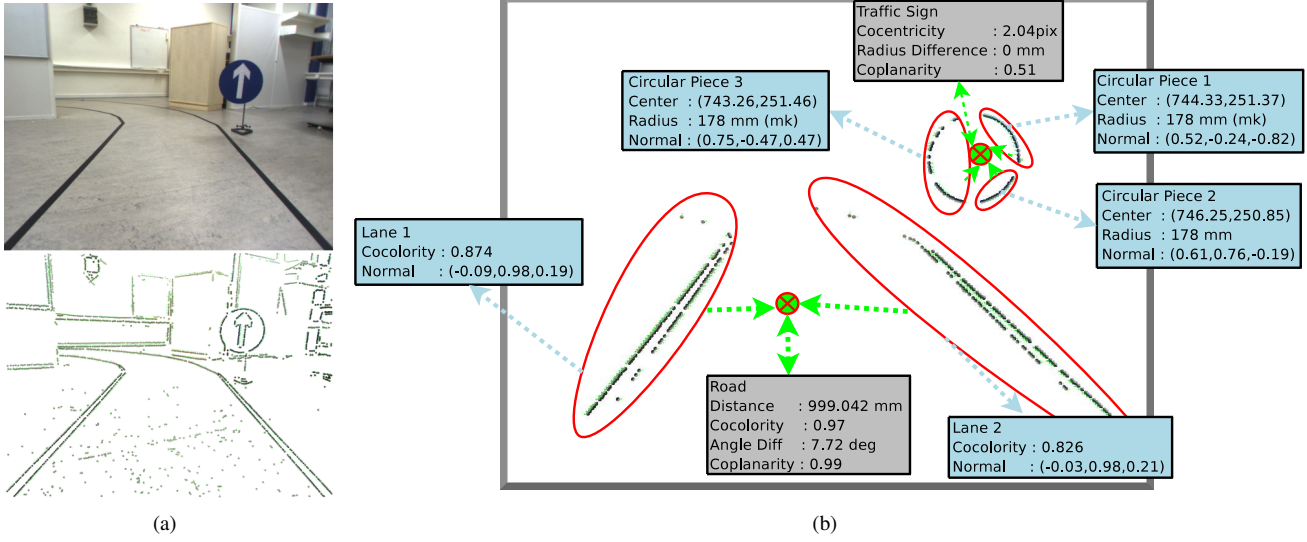


Figure 7. Interpretation of a road and a circular traffic sign. (a) Input image from a stereo pair and the corresponding 2D primitives (b) Interpretation of the scene.

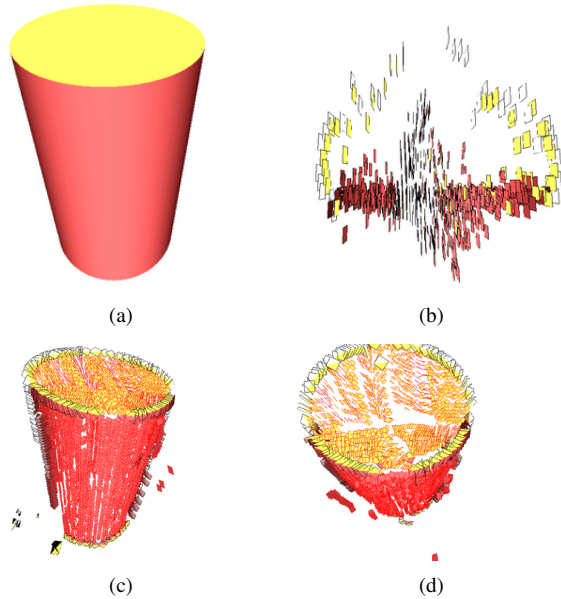


Figure 6. Depth prediction for a round object. (a) Left stereo image. (b) The top view of the results of 3D reconstruction from a dense method (taken from [17]). The dense method estimates a planar surface. The dynamic programming method from [18] produces similar results. (c)-(d) Two views of the results of our depth prediction method. Note that (b)-(d) are snapshots from our 3D visualization software.

### 3.2. Scene interpretation

Based on the co-linearity relation defined in section 2.2 we can define higher level entities, in the following called groups, as sets of co-linear primitives (for details see [16]). Although the groups of multi-modal primitives have higher

semantic meaning than individual primitives, they are not enough to define an object or give an idea about the structure of a scene. Therefore, combinations of groups are more suitable for interpreting a scene. As an example (see Figure 7), one lane of a road can be defined by a group of primitives but this group is not qualified as a road, unless it is not combined with the group that represents the opposite lane. In that sense, the opposite lane is the one that lies on the same plane with a certain distance and similar color. With a similar reasoning, a circular traffic sign is interpreted by the combination of circular pieces that shares the same center and the plane with a similar enough color.

In this way we can make use of the appearance based as well as geometric information in the primitives. Interestingly, this allows for a close to textural description of objects and scenes, e.g., the particular traffic sign in Figure 7 can be described by its geometric properties (curved and co-planar groups with a certain proximity) as well as its appearance based aspects (being blue). In this way, the introduced representations can be seen as an intermediate step towards high level representations in which by expressing the semantic relations introduced in section 2.2, abstract statements about the scene structure can be made.

### 3.3. Grasping

In [1], it has been shown how geometry, appearance and spatial relations between multi-modal features can guide early reactive grasping which is an initial "reflex-like" grasping strategy. A simple parallel jaw gripper was used and five elementary grasping actions, called EGAs, were associated to co-planar primitives. Two samples are shown in Figure 8(a). The EGAs were tested in a simulation en-



vironment [1] as well as in a real environment. It has been shown that with a rather weak assumption of co-planarity and hence without any a-priori object knowledge, successful grasps could be generated which can then be haptically verified and used further in a cognitive system (see section 3.4). Basically, plane hypotheses based on co-planar features (as discussed in section 2) become associated to grasp hypothesis (see Figure 8(b)). By making use of the additional relations co-colority and co-linearity, the number of potential grasp hypotheses could be further reduced.

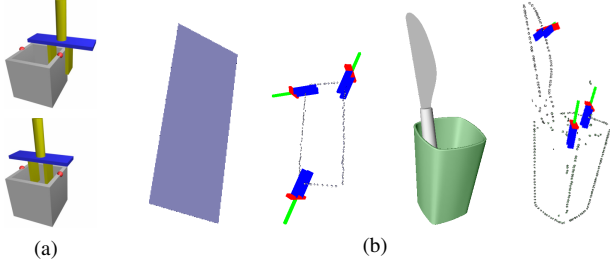


Figure 8. Sample elementary grasping actions and grasping hypothesis from [1] (a) Two sample EGAs (b) Two sample grasp hypotheses.

Even more reliable grasping hypotheses can be associated to object parts (see, e.g., [2]). To grasp cylindric or conic objects, grasping options can be associated to a circle (see Figure 10). Here, instead of using second-order relations between multi-modal primitives, 3D locations of circles have been used to generate grasping hypotheses.

To extract a 3D circle, it is important to switch between the 2D and the 3D aspects. The first step is locating the 3D circle by using the fact that a circle in 3D can be approximated by an ellipse in 2D. Although fitting an ellipse to 2D data is easier than fitting a circle in 3D, an ellipse does not give sufficient information about the center, radius and the plane normal of the 3D circle. At that point, it is possible to switch the dimension and obtain the missing information by processing the 3D features that correspond to the 2D features which form the ellipse. Fitting a plane to the 3D features determines the normal of the circle. Finally, the intersection of this plane and the line that passes from the camera center and the multiplication of the pseudo-inverse of the projection matrix and 2D ellipse center gives the center of the circle. An example of the procedure is given in Figure 9 (a-c).

Once a circle is found in 3D, four different grasp hypothesis can be generated (see Figure 10). The first one uses the center and the normal of the circle to place the gripper inside the circle and uses the radius to grasp the object from inside. For the second hypothesis, a point on the circle is calculated and this point is used to grasp the object from its brim. For the third hypothesis, the center and the normal of the circle is used for placing the gripper orthogonal to

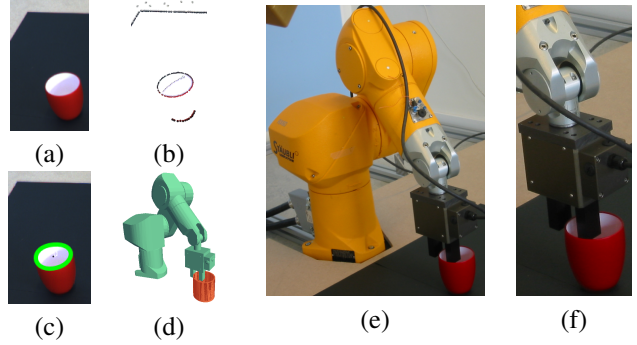


Figure 9. Grasping of a cylindrical cup (a) Input left image (b) Corresponding 2D primitives (c) Detected circle (d) Model of the robot (e-f) The cup is grasped by the robot with respect to the extracted information.

the circle normal, the radius is used to open the gripper and the object is grabbed from the side. The last hypothesis is similar to the first one but instead of inner side, the circle is grasped from outer side. A sample grasp of the second type is presented in Figure 9 (e-f).

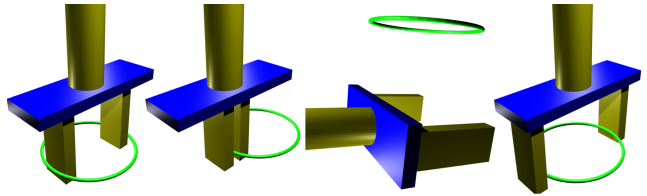


Figure 10. Four different grasp hypotheses for circles

### 3.4. Learning objectness and object shape

The detection of features belonging to one individual object is not a trivial task when a stereo system only observes a scene since there is no decision criterion that a set of features actually can be separated from the rest of the scene. However, having achieved a successful grasps (as explained in section 3.3), the robot has physical control over a potential object, and it can try to move it (see Figure 11). Since the change of primitives under a rigid-body motion can be described analytically (see section 2.2), predictions about the change of primitives can be derived. Only primitives that change according to these predictions are supposed to be part of the object.<sup>3</sup> In Figure 12, a number of representations are shown that have been extracted by this method (for details, see [14]). First steps in using these object representations for pose estimation and grasping are made in [3].

<sup>3</sup>Note that the primitives belonging to the grasper change according to the robot motion but they can be eliminated using the model of the grasper.



Figure 11. The robot is doing a rotation to extract the 3D model of a basket.



Figure 12. Sample objects and their related accumulated representation [14].

## 4. Discussion

The advantages of using a 2D or a 3D scene representation is highly dependent on the application and the context. Both have their own advantages and disadvantages as presented in Table 1. By keeping these properties in mind, we described a representation that preserves relevant aspects of 2D and 3D information to allow for switching between the dimensions according to the actual requirements. We exemplified the potential of this approach in four applications of rather different nature, covering depth estimation at homogeneous areas, semantic scene description, grasping and extraction of object representations.

## References

- [1] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Early reactive grasping with second order 3d feature relations. *IEEE International Conference on Robotics and Automation (ICRA), Workshop: From features to actions - Unifying perspectives in computational and robot vision*, 2007.
- [2] I. Biederman. Recognition by components: A theory of human image understanding. *Psychological Review*, 94(2), 1987.
- [3] R. Detry and J. Piater. Hierarchical integration of local 3d features for probabilistic pose recovery. *Robot Manipulation: Sensing and Adapting to the Real World, 2007 (Workshop at Robotics, Science and Systems)*, 2007.
- [4] S. Edelman and H. H. Bulthoff. Orientation dependence in the recognition of familiar and novel views of 3D objects. *Vision Research*, 32:2385–2400, 1992.
- [5] M. Felsberg and N. Krüger. A probabilistic definition of intrinsic dimensionality for images. *Pattern Recognition, 24th DAGM Symposium*, 2003.
- [6] W. Grimson. Surface consistency constraints in vision. *CVGIP*, 24(1):28–51, 1983.
- [7] D. Hubel and T. Wiesel. Brain mechanisms of vision. *Scientific American*, 241:130–144, 1979.
- [8] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of local 3d structure in 2d images. *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1114–1121, 2006.
- [9] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *Interdisciplinary Journal of Artificial Intelligence & the Simulation of Behaviour, AISB Journal*, 1(5):417–427, 2004.
- [10] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110, 2004.
- [11] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
- [12] J. L. Mundy, A. Liu, N. Pillow, A. Zisserman, S. Abdallah, S. Utcke, S. Nayar, and C. Rothwell. An experimental comparison of appearance and geometric model based recognition. In *Object Representation in Computer Vision*, pages 247–269, 1996.
- [13] R. Murray, Z. Li, and S. Sastry. *A mathematical introduction to Robotic Manipulation*. CRC Press, 1994.
- [14] N. Pugeault, E. Başeski, D. Kraft, F. Wörgötter, and N. Krüger. Extraction of multi-modal object representations in a robot vision system. 2007.
- [15] N. Pugeault, E. Başeski, N. Krüger, S. Kalkan, and F. Wörgötter. Reconstruction accuracy and relations. In *Signal Processing, Pattern Recognition, and Applications (SPPRA)*, submitted.
- [16] N. Pugeault, F. Wörgötter, and N. Krüger. Multi-modal scene reconstruction using perceptual grouping constraints. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR’06)*, 2006.
- [17] S. P. Sabatini, G. Gastaldi, F. Solari, J. Diaz, E. Ros, K. Pauwels, K. M. M. V. Hulle, N. Pugeault, and N. Krüger. Compact and accurate early vision processing in the harmonic space. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.
- [18] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Technical Report MSR-TR-2001-81, Microsoft Research, Microsoft Corporation, November 2001.
- [19] F. Wörgötter, N. Krüger, N. Pugeault, D. Calow, M. Lappe, K. Pauwels, M. V. Hulle, S. Tan, and A. Johnston. Early cognitive vision: Using gestalt-laws for task-dependent, active image-processing. *Natural Computing*, 3(3):293–321, 2004.

## **Appendix B**

### ***Local statistics of retinal optic flow for self-motion through natural scenes***

D. Calow and M. Lappe

*( Networks: Computations in Neural Systems – in press )*

# Local statistics of retinal optic flow for self-motion through natural sceneries

Dirk Calow and Markus Lappe

Dept. of Psychology, Westf.- Wilhelms University, Fliednerstr. 21, 48149 Münster, Germany

**Abstract** Image analysis in the visual system is well adapted to the statistics of natural scenes. Investigations of natural image statistics have so far mainly focussed on static features. The present study is dedicated to the measurement and the analysis of the statistics of optic flow generated on the retina during locomotion through natural environments. Natural locomotion includes bouncing and swaying of the head and eye movement reflexes that stabilize gaze onto interesting objects in the scene while walking. We investigate the dependencies of the local statistics of optic flow on the depth-structure of the natural environment and on the ego-motion parameters. To measure these dependencies we estimate the mutual information between correlated data sets. We analyze the results with respect to the variation of the dependencies over the visual field, since the visual motions in the optic flow vary depending on visual field position. We find that retinal flow direction and retinal speed show only minor statistical interdependencies. Retinal speed is statistically tightly connected to the depth structure of the scene. Retinal flow direction is statistically mostly driven by the relation between the direction of gaze and the direction of ego-motion. These dependencies differ at different visual field positions such that certain areas of the visual field provide more information about ego-motion and other areas provide more information about depth. The statistical properties of natural optic flow may be used to tune the performance of artificial vision systems based on human imitating behavior, and may be useful for analyzing properties of natural vision systems.

**Keywords:** optic flow, natural ego-motion, statistics, entropy estimation, mutual information estimation

## 1 Introduction

Often the brain has to analyze sensory signals which are ambiguous. Ambiguity arises from the spatial and/or temporal properties of the perceptual sensors, from noise introduced by the perceptual sensors, and from noise created by the environment. To (re)construct perception, the brain may use statistically plausible predictions and/or statistical models of the signal-sending environment. The resources of a signal processing system are usually limited, and therefore the range of signals that can be processed is bounded. Non-linear processing schemes that include knowledge of the statistics of the signals can enable the system to be more sensitive for signals which occur very frequently, and to attach less value to signals which are very unlikely to occur. Such statistically efficient processing schemes restrict the limited resources of the system to the range of statistically probable signals. Therefore, evolutionary adaptations of the perceptual areas of the brain to the statistics of



natural environments are plausible. Effects of such adaptations are seen in gestalt laws (Elder & Goldberg, 2002; Krüger & Wörgötter, 2002) and in efficient encoding schemes (Barlow, 1961; Laughlin, 1981).

In the visual modality, several researchers invested effort to reveal the statistics of natural environments, and to link it with the neural representation of the sceneries (Laughlin, 1981; Rudermann & Bialek, 1994; Atick & Redlich, 1992; Olshausen & Field, 1996; Krüger, 1998; van Hateren & Rudermann, 1998; Zetsche & Krieger, 2001; Berkes & Wiskott, 2002; Simoncelli & Olshausen, 2001; Betsch et al., 2004). Their investigations are largely restricted to static attributes of natural scenes, however, even when dynamic stimulus material was used (van Hateren & Rudermann, 1998; Betsch et al., 2004). Furthermore, the resulting statistics are treated as independent of the position in the field of view. The properties of motion signals elicited on optic detectors by ego-motion within natural sceneries strongly depend on the position in the view field (Zanker & Zeil, 2005). To investigate the statistics of these motion signals therefore requires an analysis of distributions of flow vectors with respect to their visual field position.

Optic flow generated by self motion encodes much information about the direction of ego-motion, the velocity, the distances of potential obstacles and the structure of the environment (Gibson, 1950, 1966). Animals use this information for path planning, obstacle avoidance, ego-motion control, and foreground-background segregation (see Lappe (2000b) for an overview). The motion signals of the optic flow are processed in specialized motion-processing brain areas (Albright, 1989; Saito et al., 1986; Duffy & Wurtz, 1991; Lappe et al., 1996). It is likely that the motion-processing pathway of the brain uses statistical properties of natural flow fields to efficiently encode natural optic flow, and to reconstruct the true motion field from the motion signals in early motion detectors. We hypothesize that the brain has involved mechanisms of extracting information from optic flow which benefit from statistical dependencies of the elicited optic flow on the properties of the natural environment and natural motion situations. An investigation of the local statistical properties of optic flow can be the starting point to reveal such connections.

The analysis of the statistics of optic flow may be undertaken on the true motion signals (Ivins et al., 1999; Calow et al., 2004; Roth & Black, 2005), or on the signals obtained from early motion detectors (Fermüller et al., 2001; Zanker & Zeil, 2005; Kalkan et al., 2005). The latter approaches analyze the combination of properties of the motion field generated by ego-motion with the properties of particular motion detectors. Since we are interested in the statistical properties of the motion field itself we need to analyze the true motion signals. Therefore, we need a large number of true motion fields generated by natural ego-motion through a natural environment.

A method to collect a sufficient number of true optic flow fields was introduced in Calow et al. (2004). Based on the Brown range image data base (Huang et al., 2000) true motion fields were generated by biologically plausible ego-motion and first results of the first order statistics of retinal optic flow fields were reported. Roth and Black (2005) used this method to investigate the statistics of optic flow and elementary optic flow components.

Since their work mainly focused on aspects of machine vision, the ego-motion parameters underlying the optic flow fields were obtained from ego-motions of hand-held or car mounted cameras. The resulting statistics were treated as independent of the position in the field of view. Our investigation is dedicated to the local statistics of the true retinal motion signals occurring in biological vision during natural, human-like ego-motion. In natural locomotion, eye movement reflexes stabilize gaze on objects of interest in the scene (Solomon & Cohen, 1992; Lappe et al., 1997; Niemann et al., 1999) such that natural ego-motion is always a combination of body movement and eye movement. The combination of body movement, eye movement, and depth structure of the visual environment determines the structure of the optic flow on the retina (Lappe et al., 1999). Our investigation of the statistical properties of the flow field is therefore based on a combination of walking and eye-movement reflexes.

We use the term local statistics to note the statistical properties of the distributions of retinal velocities and their statistical dependencies on depth and ego-motion for certain positions in the field of view. The correlations between motion signals of different positions are not part of our notion of local statistics.

We see the purpose of our study in providing basic information and quantitative data on the statistics of retinal motion signals. This information can be used to predict sensitivity ranges of neurons in the motion processing pathway of the brain. Future work will focus on the examination of the hypothesis that these neurons efficiently encode distributions of naturally occurring retinal motion signals. The knowledge of the statistics is crucial for that purpose. Furthermore, the knowledge of the statistical properties of retinal motion signal is an important tool in creating experimental paradigms that focus on natural motion stimuli. Comparisons between natural and unnatural motion situations are necessary to reveal how the motion processing pathway is adapted to the statistics of the natural environment. Our investigation can also provide prior knowledge for creating probabilistic models of the motion processing pathway of the brain based on Bayesian inference (Weiss & Fleet, 2001).

Since the local statistics of optic flow are tightly linked to the statistics of the depth structure of natural scenes and to the statistics of the ego-motion parameters the information about the depth map of the current scene and the ego-motion situation is encoded in the retinal flow. However, the generation of optic flow maps from a five dimensional parameter space (walking speed, heading, depth, and depth of the fixation point) to a two dimensional flow vector (cf. equations (5) and (6)). Therefore, the information about the underlying parameters is condensed in the flow vector and cannot be extracted from an individual flow vector directly. Recovery of heading, for instance, requires the combined information from several flow vectors (Longuet-Higgins & Prazdny, 1980). However, different areas in the field of view show different statistical dependencies of the components of the optic flow on heading and depth. By focussing the analysis on these areas the brain may gain instant access to particular parameters regarding the other parameters as fixed and their variation as noise.

Our analysis starts with the measure of dependence between the random variables retinal speed and direction for a set of positions in the field of view. Then we analyze the properties mean, standard deviation, skewness, kurtosis, and negentropy of the distributions of speed

and direction. The results provide the most important properties of the distributions depending on the position in field of view. Finally, we investigate the statistical dependencies between the distributions of optic flow and the distributions of depth in the scene, depth of the fixation point, and heading. To put the influence of heading and scene structure into context, the same analysis is performed with two other sets of optic flow fields. One set is generated under the assumption that no gaze stabilization is executed and therefore no rotation occurs. The second set provides a baseline for comparison of the influence of the scene structure. In this set, the depth values are randomly mixed. Thus, the scene structure is abolished but the depth statistics do not differ for different positions.

## 2 Methods

### 2.1 Construction of retinal flow fields

In this section, we describe the preparation of retinal flow fields in a sufficient number for the statistical analysis. The calculation of retinal optic flow fields relies on the knowledge of the depth map of a variety of natural scenes. We will explain how to obtain ego-motion parameters and how to construct flow fields from the depth map and the ego-motion.

We generate flow fields under three different conditions. One condition is regarded as naturalistic and combines naturalistic ego-motion, which includes gaze stabilization, through natural scenes (natural condition). Another set of flow fields relies on the same set of natural scenes and heading directions but without gaze stabilization (non-stabilized condition). In this set, gaze is directed to the same objects in the visual field as in the natural condition but is not stabilized on that object, i.e. does not counteract the motion induced by the forward movement. The third set of flow fields is generated from the same naturalistic ego-motion parameters, including gaze stabilization, but each scene is mixed in depth by exchanging the depth values between randomly selected pairs of positions (mixed depth condition). This procedure ensures that the overall distribution of depth values is natural, but the differences in depth statistics for different positions in the visual field disappears.

#### 2.1.1 Database.

We use the Brown Range Image Database, a database of 197 range images collected by Ann Lee, Jinggang Huang and David Mumford at Brown University (Huang et al., 2000). The range images are recorded with a laser range-finder with high spatial resolution. Each image contains  $444 \times 1440$  measurements with an angular separation of 0.18 degree. The field of view is 80 degree vertically and 259 degree horizontally. The distance of each point is calculated from the time of flight of the laser beam, where the operational range of the sensor is 2 – 200m. The wavelength of the laser beam is  $0.9\mu m$  and lies in the near infrared

region. Thus, the data of each point consist of 4 values: the distance  $R$ , the azimuth angle  $\phi$ , the zenith angle  $\theta$ , and a value for the reflected intensity of the laser beam. The location of the source of the laser beam is  $1.5m$  above the ground. Figure 1 shows a typical range-image projected onto the  $\phi - \theta$  plane. It can be seen that the intensity of the reflected laser beam characterizes the properties of the reflecting surfaces sufficiently well. The objects in the scene are clearly visible, and the image resembles a grey level picture of a fully illuminated scene at night. The data are provided in spherical coordinates  $R, \phi, \theta$ . The three dimensional Euclidian coordinates from the standpoint of the laser range finder can be easily calculated by  $(X, Y, Z) = (R \cos(\phi) \sin(\theta), R \sin(\phi) \sin(\theta), R \cos(\theta))$ .

### 2.1.2 Retinal projection.

The knowledge of the 3D coordinates of each image point allows the calculation of the true motion of that point for any given combination of translation and rotation of the projection surface. As we are interested in the statistics of retinal projections, we consider as the retina a spherical projection surface with the radius 1. All coordinate systems we will use in the following are attached to the center of the projection surface and therefore the coordinates of the data delivered by the data base have to be transformed in the perspective of the projection surface. In Euclidian coordinates, the  $X$ -and  $Y$ -axis are right and up, and the  $Z$ -axis is perpendicular to the  $X$ - $Y$  plane. The value of the  $Z$  coordinate of any point in the scene is the depth of that point from the perspective of the projection surface. The most simple description of optic flow vectors on the sphere is given by the following notation. Let  $\epsilon$  be the angle of eccentricity describing the meridians of the sphere and  $\sigma$  the rotation angle describing the circles of latitude rotating counter clockwise. The focal point is defined by  $\epsilon = 0$ . The meridians and the circles of latitude are coordinate lines, and every vector  $v$  on the sphere has the components  $v = (v_\epsilon, v_\sigma)$  in the respective local orthonormal coordinate system. The velocity  $v$  of a point moving over the sphere described in terms of the temporal derivatives of  $\epsilon$  and  $\sigma$  is  $v = (\frac{d\epsilon}{dt}, \sin(\epsilon) \frac{d\sigma}{dt})$ . Although the spherical coordinates  $\epsilon, \sigma$  already sufficiently provide the description of the sphere, we want to use a second spherical coordinate system to denote positions on the projection surface, in terms of which we are going to plot our results. Each position on the sphere is described by the azimuth  $\tilde{\phi}$  and the elevation  $\tilde{\theta}$ , where the projection of a point in the scene onto the sphere is governed by the relationship in equation

$$\begin{aligned} \frac{X}{Z} &= \frac{\sin(\tilde{\phi})}{\cos(\tilde{\phi})} = \frac{\sin(\epsilon)}{\cos(\epsilon)} \cos(\sigma) \\ \frac{Y}{Z} &= \frac{\sin(\tilde{\theta})}{\cos(\tilde{\phi}) \cos(\tilde{\theta})} = \frac{\sin(\epsilon)}{\cos(\epsilon)} \sin(\sigma). \end{aligned}$$

Since, positions of the upper and the right visual field are denoted by positive values of  $\tilde{\theta}$  and  $\tilde{\phi}$  respectively and positions of the lower and left visual field are denoted by negative values of  $\tilde{\theta}$  and  $\tilde{\phi}$  respectively, the reader can easily discern what positions on the sphere are pointing to the right, left, up and down from the perspective of the observer. The flow field emerging

on a moving sphere can be easily extracted by a simple transformation from the well known flow field  $(v_x, v_y)$ , which would be generated on a moving plane with internal coordinates  $(x, y) = (X/Z, Y/Z)$  (Longuet-Higgins & Prazdny, 1980). The flow field generated on a plane is described by

$$\frac{dx}{dt} = v_x = \frac{1}{Z}(-T_x + xT_z) + (xy\Omega_x - (1+x^2)\Omega_y + y\Omega_z) \quad (1)$$

$$\frac{dy}{dt} = v_y = \frac{1}{Z}(-T_y + yT_z) + (-xy\Omega_y + (1+y^2)\Omega_x - x\Omega_z). \quad (2)$$

The transformation rule is

$$v_\epsilon = \cos^2(\epsilon)(\cos(\sigma)v_x + \sin(\sigma)v_y) \quad (3)$$

$$v_\sigma = \cos(\epsilon)(\cos(\sigma)v_y - \sin(\sigma)v_x). \quad (4)$$

### 2.1.3 Ego-motion parameters.

To calculate the flow field from the scene structure we need the motion parameters of the projection surface. The ego-motion of the surface is fully described by the translational velocity vector of the surface  $T = (T_x, T_y, T_z)$  and the vector of rotation  $\Omega = (\Omega_x, \Omega_y, \Omega_z)$  in the Euclidian coordinate system attached to the projection surface. The translation  $T$  of the surface can be further split up in the parameters translational or walking speed  $\|T\|$  and heading  $(H_\phi, H_\theta)$ , which are azimuth and elevation denoting the direction of the translational velocity vector of the surface:

$$T = (T_x, T_y, T_z) = \|T\|(\cos(H_\phi) \sin(H_\theta), \cos(H_\theta), \sin(H_\phi) \sin(H_\theta)).$$

Natural ego-motion within the scenes involves eye movements which stabilize the gaze on environmental objects (Lappe, 2000a; Lappe et al., 1998). Gaze stabilization keeps the point of interest or the gaze attracting object in the center of the visual field and causes the motion in the center of view to be zero. It can be easily extracted from equations (1) and (2) that the associated rotation depends on the translation by  $\Omega = \frac{1}{Z_f}(Ty, -Tx, 0)$ , where  $Z_f$  denotes the depth of the point at which gaze is directed. Under this assumption (3) and (4) can be transformed to

$$v_\epsilon = \frac{1}{Z} \left( \cos(\epsilon) \sin(\epsilon) T_z + \left( \frac{Z}{Z_f} - \cos^2(\epsilon) \right) (\cos(\sigma) T_x + \sin(\sigma) T_y) \right) \quad (5)$$

$$v_\sigma = \frac{1}{Z} \cos(\epsilon) \left( \frac{Z}{Z_f} - 1 \right) (\cos(\sigma) T_y - \sin(\sigma) T_x). \quad (6)$$

By (5) and (6) the parameters governing the optic flow at a certain position are the walking speed  $\|T\|$ , the heading  $(H_\phi, H_\theta)$  and the depth-structure of the scene determined by the depth  $Z$  of the point in question and the depth of the fixation point  $Z_f$ .

For the condition without gaze stabilization the concerning retinal flow can be extracted from (5) and (6) by assuming the observer gazes towards a point in infinity, i.e.  $Z_f \rightarrow \infty$ , and thus  $\Omega$  and the term  $Z/Z_f$  vanish.

Since, for higher walking speed the distribution is linearly shifted to higher speed values, there are only trivial correlations between the motion signals and walking speed. Therefore, we restrict our analysis to flow fields generated by a walking speed of  $\|T\| = 1.4$  meter per second.

Finding plausible ego-motion parameters ( $H_\phi, H_\theta$ ) and depth of fixations  $Z_f$  for the respective scene requires to search for feasible walking directions within the scene and to extract probable gaze directions. The walking direction within the scene combined with the gaze direction provides the parameters of heading ( $H_\phi, H_\theta$ ). Furthermore, if the direction of fixation is given, the depth of fixation can be extracted from the point in the laser range image that the gaze direction is associated to.

To determine possible walking directions within a range image we search for areas which are free from obstacles in a depth of at least  $3m$  and a width of  $0.7m$ . This criterion gives us a set of walking directions for each scene, which are considered to be equally likely.

To obtain gaze directions that we can use to generate gaze stabilization movements we measured eye movements of observers who viewed images, which were generated from segments of the range images centered on the walking directions. The images are projected onto a  $36.5cm \times 27.5cm$  plane with a focal length of  $30cm$  (white frame in Figure 1 A). Six subjects viewed these pictures on a 17 inch computer monitor with the head stabilized on a chin rest  $30cm$  in front of the monitor. Pictures were shown for 1 second in immediate succession to give the impression of a changing environment the subject is moving through. Gaze fixation points were measured by an eye tracking system (Eye Link II). The first fixation for each picture was rejected because it might be partially driven from the preceding picture. The subsequent fixations were used as probable gaze directions for the statistical analysis. Although, the subjects are not actually walking through the real scenes, and therefore have no access to the true color, disparity and other factors, which might influence gaze attraction, the arrangement of objects and surfaces populating the scene and the objects itself are well recognizable (Figure 1 A). Furthermore, humans are usually familiar with that sort of scenes and this world knowledge ensures that the scenes are instantly identifiable as street scenes or forest scenes, and that gaze is instantly attracted to the usual objects in such scenes. Figure 1 B shows the distribution of gaze directions while viewing the scenes. The points are plotted in spherical coordinates  $(\phi, \theta)$  and are centered on the direction of walking used for the flow field calculations.

To consider all aspects of human walking we also take bouncing and swaying of the head during walking into account. Bouncing and swaying of the head while walking is part of the complex oscillatory motion pattern of during walking (Imai et al., 2001). The position of the head during walking can be modeled as sinusoidal time functions. To obtain typical values for the amplitude and period we measure the head position of one human subject while walking. The walking velocity of the subject was 1.4 meter per second. The height of the subject was 1.80 meter. The head-position was measured by a position tracking system (Motion Star). We approximate the properties of vertical and horizontal movement of the head as follows. The vertical head position has a period of 0.6 second and an amplitude of

0.02 meter. The horizontal head position has a period of 1.2 second and an amplitude of 0.02 meter as well. The velocity of the head during walking can be obtained by the first temporal derivative of the horizontal and vertical head position. The actual horizontal and vertical movement of the head for a certain motion situation is picked up at a randomly selected time and is added to the preliminary determined ego-motion parameters such that the gaze of the eye towards the fixation point is stabilized also during bouncing and swaying of the head.

Since the actual combination of walking direction, gaze direction and head movement is linked to the environment and to the task, and since the simulated components of ego-motion might not be generally independent, our assumed ego-motion is an approximation to actual ego-motion and might not match actual ego-motion in all details. But the simulation matches the main components of human ego-motion and allows us to combine naturalistic ego-motion parameters with the true depth information data provided by the range image database.

We mirror each scene and the respective heading on the vertical plane (Y-Z plane). This procedure of mirroring the scene attaches to each position in the field of view the depth value of its counterpart in the opposite hemisphere and therefore doubles the set of depth data points for each position. Simultaneously mirroring the heading ensures that ego motion is still in the direction of the obstacle free corridor.

#### 2.1.4 The retina

The flow fields we finally consider are elicited on the inside of a section of a sphere, in which a grid of motion sensors is affixed. The field of view of this retina is set to  $90^\circ$  horizontally and  $58^\circ$  vertically. This field of view is subdivided in pixels, which can be referred to as motion sensors, with a resolution of  $0.36^\circ \times 0.36^\circ$  yielding a grid of  $250 \times 160$  pixels. As the angular separation of the range images is  $0.18^\circ$ , one pixel covers up to 4 data points. The depth values  $Z = R \sin(\phi) \sin(\theta)$  from these data points are averaged. The mean depth value is assigned to the pixel in question. Thus, we reduce the original resolution provided by the laser range images. This procedure is necessary, because the pixel grid of the retina is sliding over the pixel grid of the laser range image and mostly does not match the original pixels. Therefore, reducing the resolution ensures that all pixels of the retina receive appropriate motions signals. The flow vector attached to a pixel depends on the depth value, the translation and rotation components of the ego-motion and the visual field position of the pixel. The flow vectors of all pixels of an image provide the measurement of the true retinal flow field for this gaze direction, ego-motion, and scene. Figure 1 C shows an example of a true retinal flow field.

## 2.2 Statistical analysis

We constructed 7136 different flow fields for each of the three conditions naturalistic, no gaze stabilization, and mixed depth map. To examine the local statistics of these flow fields we collect for each scene, motion situation, and pixel position the data sets which comprises the retinal velocity, the depth at the respective position, the depth of fixation, and the heading of the respective ego-motion. Examples of the distributions of retinal velocities can be seen in Figure 2.

The analysis of the local statistics of optic flow is divided into two parts. The first part is dedicated to the examination of the distributions of optic flow and how strong the statistical properties of which depend on the positions in the field of view. The investigation considers the polar optic flow components retinal flow direction and retinal speed and starts with the measure of the statistical dependence of these two variables. Although there are slight differences in the degree of statistical dependence for the different conditions, it turns out that flow direction and retinal speed are largely statistically independent for all conditions. Therefore, the further analysis is based on the extracted one-dimensional distributions of retinal speed and retinal flow direction. For each position in the field of view we measure the mean, the scatter, the skewness, the kurtosis and an estimation of the negentropy for the distributions of retinal speed and retinal flow direction.

The second part addresses the problem of the statistical dependence of the variables retinal direction and retinal speed on the particular parameters depth, depth of fixation point, and heading. The statistical correlations are not purely linear, and nonlinear statistical dependencies play an important role. This can be seen in the exemplary scatter-plots (Figure 3), which show the extracted data for the distribution of retinal speed and the inverse of depth (Figure 3 A), the retinal speed and the elevation heading component (Figure 3 B), and the retinal speed and the azimuth heading component (Figure 3 C) at the visual field position  $(-5^\circ, -18^\circ)$ . All scatter plots show that a statistical analysis based on linear correlation is not sufficient. For example in the scatter plots Figure 3 A and C a quadratic correlation seems to underlie the data set and in the scatter plot Figure 3 B a correlation of degree three provides the main contribution. However, despite the very different kinds of statistical dependencies for different data sets and for different positions we would like to compare the degree of dependence the retinal flow has on the different parameters. Information theory provides the notion of mutual information between random variables. The mutual information is a measure of the difference between the joint probability density function (PDF) of the random variables and the PDF which would appear if the random variables were statistically independent. Since the mutual information only vanishes if the random variables are fully statistically independent, the measure of the mutual information takes all kinds of statistical dependencies into account and is useful to investigate the degree of dependence between random variables. However, the mutual information can range from zero to infinity. Therefore, we are going to define a generalized dependence coefficient in terms of the mutual information between data sets which will be bound between values 0 and 1. This definition is motivated by the mathematical relation between mutual information



and the linear correlation in the case of purely linear statistical dependency.

### 2.2.1 Polar representation of retinal flow vectors

The retinal velocity at a certain position in the field of view is a two dimensional vector  $(v_\epsilon, v_\sigma)$ . Therefore, the velocity distributions are distributions of two dimensional random variables. The random variables we choose for further analysis are the polar coordinates retinal speed  $v$  and retinal direction  $\phi_{dir}$ . We will show that these random variables are largely independent. Thus, our analysis of the local statistics of retinal velocities will be separately performed on these two random variables:

$$\begin{aligned} v &= \sqrt{v_\epsilon^2 + v_\sigma^2}, \\ \phi_{dir} &= \begin{cases} \arccos\left(\frac{v_\epsilon}{\sqrt{v_\epsilon^2 + v_\sigma^2}}\right) & ; \quad v_\sigma \geq 0, \\ -\arccos\left(\frac{v_\epsilon}{\sqrt{v_\epsilon^2 + v_\sigma^2}}\right) & ; \quad v_\sigma < 0. \end{cases} \end{aligned}$$

### 2.2.2 Estimating the properties of the distributions of speed and direction

To illustrate the different properties of the distributions of speed and direction for different positions in the visual field we measure the mean ( $EX$ ), the scatter ( $DX$ ), the skewness ( $M_3X$ ), the kurtosis ( $M_4X$ ), and estimate the negentropy ( $JX$ ). For the sake of completeness, we list the well known corresponding formulas to estimate these parameters from an empirical data set  $X = \{x_i \in \mathbb{R}\}_{i=1,2,\dots,N}$ :

$$\begin{aligned} EX &= \frac{1}{N} \sum_{i=1}^N x_i \\ DX &= \sqrt{M_2X} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - EX)^2} \\ M_3X &= \frac{\frac{1}{N-1} \sum_{i=1}^N (x_i - EX)^3}{DX^3} \\ M_4X &= \frac{\frac{1}{N-1} \sum_{i=1}^N (x_i - EX)^4}{DX^4} - 3. \end{aligned}$$

$DX$  is a measure for the width or the spreading of the data,  $M_3X$  and  $M_4X$  are measures of the difference between the empirical distribution and the Gaussian distribution.  $M_3X$  measures the asymmetry and  $M_4X$  is a measure for the flatness of the distribution. A negative value of  $M_4X$  means the distribution is more flat than a Gaussian and has shorter tails. A positive value of  $M_4X$  means the distribution has a peak higher than a Gaussian and longer tails.

Mean, scatter, skewness and kurtosis take only the first four moments of a distribution into account. To obtain a compact measure to assess the difference between an empirical distribution and the Gaussian distribution with the same mean and scatter we estimate the negentropy from the empirical distribution. Let  $P(x)$  be the probability density function (PDF) of a random variable  $X$ . The negentropy  $J(X)$  is defined as the difference between the differential entropy of the Gaussian  $H(X_{gauss})$  and the actual differential entropy  $H(X) := - \int_{supp P} P(x) \log_2(P(x)) dx$

$$J(X) = H(X_{gauss}) - H(X) = \frac{1}{2} \log_2(e 2\pi D X^2) - H(X),$$

where  $e$  is the Euler number. Since a Gaussian distribution has the maximal entropy for a given mean and scatter,  $J(X)$  is always nonnegative and vanishes only if  $X$  is a Gaussian distribution. The estimation of the negentropy of a distribution requires the estimation of the differential entropy.

### 2.2.3 The measure of dependence and the estimation of differential entropy

The analysis of the statistical interdependence between the optic flow components retinal direction and retinal speed and the statistical dependencies of the optic flow components on the parameters depth, depth of fixation point and heading components requires estimating the mutual information from two-dimensional and three-dimensional data sets. The analysis of the properties of the distributions of retinal speed and retinal direction furthermore needs an estimation of the differential entropy from one-dimensional data sets.

The mutual information  $mI(X_1, X_2, \dots, X_n)$  between (possibly more-dimensional) continuous random variables  $X_i$  with minor PDF's  $P_i(x_i)$  and joint PDF  $P_X(x_1, x_2, \dots)$  is defined by

$$\begin{aligned} mI(X_1, X_2, \dots) &= \int_{\mathbb{R}^m} P_X(x_1, x_2, \dots) \log_2 \left( \frac{P_X(x_1, x_2, \dots)}{P_1(x_1)P_2(x_2)\dots P_n(x_n)} \right) d^m x, \\ &= \sum_i^n H(X_i) - H(X_1, X_2, \dots), \end{aligned} \quad (7)$$

where  $m = \sum_{i=1}^n \dim(X_i)$  and

$$H(X) = - \int P_X(x) \log_2(P_X(x)) d^m x; \quad m = \dim(X) \quad (8)$$

is the (differential) entropy for the random variable  $X$ . The mutual information is always nonnegative, and zero if and only if  $P(x_1, x_2, \dots) = P_1(x_1)P_2(x_2)\dots P_n(x_n)$ , i.e. the  $X_i$  are mutually independent random variables. In this study, we deploy the method of the k-nearest neighbors distances to estimate the differential entropy (Kozachenko & Leonenko, 1987) and the modification to estimate mutual information (Kraskov et al., 2004). Let

$\Psi = \{\psi_i\}_{i=1,2,\dots,N}$  be a  $m$ -dimensional data-set and let  $\hat{P}_\Psi$  be an estimator for the actual PDF  $P_\Psi$ . The differential entropy (8) can be estimated by

$$H(\Psi) \approx \hat{H}(\Psi) = -\frac{1}{N} \sum_{i=1}^N \log_2(\hat{P}_\Psi(\psi_i)). \quad (9)$$

Let  $\epsilon_i$  be the minimal radius of the sphere centered at  $\psi_i \in \Psi$  within the  $k$  nearest neighbors of  $\psi_i$  are located, for large  $N$  and large  $k$  (but  $k \ll N$ ),  $P_\Psi(\psi_i)$  can be estimated by

$$\hat{P}_\Psi(\psi_i) = \frac{1}{V_u \epsilon_i^m} \frac{k}{N}, \quad (10)$$

where  $V_u$  is the volume of the unit ball. Equation (10) leads directly to an estimate of the differential entropy

$$\hat{H}(\Psi) = \log_2(N) - \log_2(k) + \log_2(V_u) + \frac{m}{N} \sum_{i=1}^N \log_2(\epsilon_i). \quad (11)$$

For smaller  $N$  and/or smaller  $k$  equation (10) gives a rather bad estimate of  $P_\Psi$  and equation (11) must be replaced by

$$\hat{H}(\Psi) = (\psi(N) - \psi(k)) / \log(2) + \log_2(V_u) + \frac{m}{N} \sum_{i=1}^N \log_2(\epsilon_i),$$

where  $\psi$  is the digamma function (see Kraskov et al. (2004)). Let now  $\Psi = (\Psi_1, \Psi_2) = \{(\psi_{1i}, \psi_{2i})\}_{i=1,2,\dots,N}$  be a  $(m_1 + m_2)$ -dimensional data set for which we wish to estimate the mutual information  $mI(\Psi_1, \Psi_2) \approx \hat{H}(\Psi_1) + \hat{H}(\Psi_2) - \hat{H}(\Psi)$ . Whereas  $\hat{H}(\Psi)$  can be estimated by equation (11), to use the same distance scales in the joint and the minor spaces and to avoid any biases the estimation of the differential entropies  $\hat{H}(\Psi_1)$  and  $\hat{H}(\Psi_2)$  have to be modified in the following way (see Kraskov et al. (2004)). Let  $\epsilon_i$  be the minimal radius of the sphere which is centered at  $\psi_i \in \Psi$  and which within the  $k$  nearest neighbors of  $\psi_i$  are located, then  $k_{1i}$  is the number of data located within the sphere with radius  $\epsilon_i$  centered at  $\psi_{1i}$  in the space  $\Psi_1$  and  $k_{2i}$  is the analog number of data around  $\psi_{2i}$ . The estimation of the differential entropy  $\hat{H}(\Psi_1)$  is modified by

$$\hat{H}(\Psi_1) = \log_2(N) + \frac{-1}{N} \sum_{i=1}^N \log_2(k_{1i} + 1) + \log_2(V_{1u}) + \frac{m_1}{N} \sum_{i=1}^N \log_2(\epsilon_i).$$

Analogous calculations are performed for the estimation of  $\hat{H}(\Psi_2)$ . For the estimation of mutual information between more than two random variables the method can be easily extended.

For all mutual information estimation performed in this study we fixed the number of nearest neighbors by  $k = 0.005N$ . Whereas  $N$  ranges between 6000 and 7136, the number of nearest neighbors takes values between 30 and 35.

**Rank ordering of data sets** The dependent components of an empirical data set are usually measured in different units and have different scales. Large differences in scale can cause errors in the estimation of mutual information. Note that the mutual information (7) is preserved under any differentiable transformation  $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$  of the  $m$ -dimensional components. To conform the scales of the components of the data sets the components are transformed to a uniform distribution by rank ordering.

Let  $P_X(x)$ ,  $x \in \Omega \subseteq \mathbb{R}$  be the PDF of a one-dimensional random variable  $X$ . Let  $H$  be the Heaviside step function. The transformation which turns  $X$  into a uniform random variable is

$$f_X(x) := \int_{\Omega} H(x - \tilde{x}) P(\tilde{x}) d\tilde{x}. \quad (12)$$

Let  $\{\psi_i \in \mathbb{R}\}_{i=1,2,\dots,N}$  be a one dimensional data set. Then (12) leads directly to the approximation of the uniforming procedure by

$$f_{\Psi}(\psi_i) := \frac{1}{N} \sum_{j=1}^N H(\psi_i - \psi_j), \quad (13)$$

which is referred to as rank ordering. To perform the uniforming procedure for a two-dimensional data set the first component is rank ordered according to equation (13). The resulting data set can then be divided into stripes of the same width comprising the same number of data. Regarding each stripe as a one-dimensional data set, the stripes are rank ordered by equation (13) again. Although uniforming dissolves the internal dependence structure of the two dimensional random variable, the mutual information between the uniformed two dimensional random variable and a separately rank ordered third random variable is not affected.

**The definition of the generalized dependence coefficient** Suppose the random variables  $X$  and  $Y$  have the following joint PDF:

$$P(x, y) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{x^2}{2\sigma_1^2}\right) \exp\left(-\frac{(x-y)^2}{2\sigma_2^2}\right), \quad (14)$$

and  $P_X$  and  $P_Y$  are the PDFs of the constituents. There is a simple relationship between the linear correlation coefficient  $C(X, Y)$  and the mutual information  $mI(X, Y) = \int_{\text{supp} P} P(x, y) \log_2 \left( \frac{P(x, y)}{P_X(x)P_Y(y)} \right) dx dy$ :

$$C(X, Y)^2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \quad (15)$$

$$mI(X, Y) = \frac{1}{2} \log_2 \left( \frac{\sigma_1^2 + \sigma_2^2}{\sigma_2^2} \right) \quad (16)$$

$$C(X, Y)^2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} = 1 - 2^{(-2mI(X, Y))}. \quad (17)$$

Motivated by equation (17) we define for a multi-dimensional random variable  $\Psi$  a normed mutual information

$$mI^{normed}(\Psi) := 1 - 2^{(-2mI(\Psi))}, \quad (18)$$

which takes values between 0 and 1, and which is referred to as a generalized dependence coefficient. Recall that equation (17) is only valid for linear correlations and if all constituent random variables are gaussian-distributed. However, we will use the definition (18) to condense the estimated mutual information in a value between 0 and 1 for a more compact presentation.

## 3 Results

### 3.1 Statistical interdependencies between retinal direction and retinal speed

Figure 4 shows the estimated normed mutual informations  $mI^{normed}(\Phi_{dir}, V)$  between the distributions of direction and speed for all positions in the field of view in each of the three conditions. The values of  $mI^{normed}(\Phi_{dir}, V)$  for the natural condition (Figure 4 A) range from 0.04 to 0.19, with the peak in the center of the visual field. These values are rather low suggesting that direction and speed are largely independent from each other at all positions in the visual field. This result, however, is not a direct consequence of equations (3), (4), (5), and (6) but rather depends on the statistical properties of the motion and depth parameters and their combination in walking and gaze stabilization. This can be seen from the comparison with the other two conditions.

In the condition with no gaze stabilization (Figure 4 B),  $mI^{normed}(\Phi_{dir}, V)$  ranges from 0.05 to 0.3 in the lower visual field and from 0.04 to 0.2 in the upper visual field. The interdependence between retinal speed and direction is increased for a domain of the lower visual field right under the horizontal line. This shows that the ego-motion situation influences the dependence structure of retinal speed and direction.

In the third condition (mixed depth map, Figure 4 C),  $mI^{normed}(\Phi_{dir}, V)$  varies between 0.08 and 0.16 accross the visual field. Thus, randomization of the depth structure keeps the statistical interdependence between retinal speed and direction on the same level as in the natural condition. Thus, the depth structure exerts less influence on the interdependence between retinal speed and direction than the ego-motion situation. However, a depth structure with very different statistics may affect the interdependence between direction and speed. For instance, if the scene contains only objects in great distances from the observer the retinal motion signals are mostly caused by the rotational component of ego-motion. This produces a higher statistical interdependence between direction and speed. Therefore, the low statistical interdependence between direction and speed of the optic flow in the natural condition is a particular property of ego-motion through natural settings.

The statistical interdependence between retinal direction and speed is not dependent on walking speed. Variation of walking speed only scales the retinal speed by a proportionality factor. However, a statistical variation of walking speed between different motion situations would evidently further diminish the level of statistical interdependence between retinal direction and speed by introducing an additional statistical variance which solely affects the statistics of retinal speed.

In the remainder of our analysis we consider the statistical properties of the distributions of retinal direction and speed separately. This is justified by the low statistical interdependence between direction and speed in the natural condition and facilitates the understanding and interpretation of the results.

## 3.2 Properties of the distributions of speed and directions

Figure 5 shows some examples for kernel-based estimates of the PDFs of direction (measured as deviations from the radial direction) and speed (Parzen, 1962; Silverman, 1986). All examples are from the natural condition. They show different positions in the left visual field. The distributions of the right visual field are essentially mirror-symmetric. The estimated distributions of speed appear similar to logarithmic Gaussian distributions. We therefore measure the skewness, kurtosis and negentropy for the logarithmic values of speed rather than for the speed itself to reveal how close the speed distributions are to logarithmic Gaussian distributions. We note the reference to the logarithm of a random variable by the prefix log (for example log-kurtosis).

### 3.2.1 Properties of the distributions of directions

Figure 7 shows the visual field maps for mean, scatter, skewness, kurtosis, and negentropy for the distributions of direction in the three conditions. First, we discuss the results concerning the natural condition (column A). The top panel (Figure 7 A1) shows the mean of the distributions of direction for all positions in the field of view. The mean deviates from the radial direction by up to 12 degrees. The deviation from radial is high near the center of the field of view and decreases towards the periphery. The variations of the deviation with eccentricity and the absolute values of deviation are very similar in the upper and lower visual fields. The visible cloverleaf structure in the plot shows that the means of the direction in each quadrant are distorted towards the vertical direction: the means are negative in the left upper visual field and positive in the left lower visual field, and vice versa for the right visual field. A similar structure occurs in the map of the skewness of the direction distributions (Figure 7 A3). This means that the direction distributions have longer tails at the side where the directions are closer to the vertical direction. Consequently, the skewness vanishes for the positions on the horizontal and vertical meridian.

The shifts of the means and the distortions of the distributions are mainly caused by the interplay between the mathematics of the projection and the properties of the distribution of headings. The distribution of headings has a higher variance for the horizontal component than for the vertical component (cf Figure 1 B). When heading is varied symmetrically around the center along the horizontal meridian the flow vectors induced at position along the 45 degree diagonal in the lower visual field are distributed asymmetrically around the radial direction (Figure 6). As our flow fields include eye rotations to stabilize gaze on an attended object in the scene, the flow vectors are additionally influenced by the properties of the distributions of  $\frac{Z}{Z_f}$  (see Equations (5) and (6)).  $\frac{Z}{Z_f}$  takes values between 0 and infinity. The distribution of  $\frac{Z}{Z_f}$  is right skewed. The result is a further skewing of the direction distributions. Therefore, the extent to which the resulting distributions of flow direction are skewed depends on the interplay between the statistics of the depth structure and the statistics of the ego-motion parameters. The different depth statistics in the upper and lower visual fields lead to the differences in the magnitude of skewness in the upper and the lower visual field in Figure 7 A3.

Figure 7 A2 shows the scatter of the directions around the mean. The scatter is maximal (about 60 degrees) at the center of the field of view and decrease to around 10 degrees in the periphery. In combination with the mean the scatter map shows that the direction distributions in the periphery become more radial. There is not much difference between the upper and the lower visual field.

The plots of kurtosis and estimated negentropy (Figure 7 A4 and A5) show that in large areas of the lower visual field the kurtosis and negentropy are very small (from -0.4 to 2.0 and 0.02 to 0.08 respectively). Kurtosis and negentropy increase (up to 17 and 1 respectively) near the horizontal meridian. However, also comparatively small deviations from zero kurtosis and from zero negentropy, such as those in the lower visual field, give a significant difference of the distribution from a Gaussian. For example, the distribution at position  $(-30, -15)$  in Figure 5 A has a kurtosis of 0.44, a negentropy of 0.05, and a skewness of 0.55, and is clearly different from a Gaussian distribution. Position  $(-30, 15)$  in Figure 5 A provides an example for a distribution with rather large values of kurtosis (6.54), negentropy (0.28), and skewness (-1.28).

We conclude that the distributions of the directions for positions of the lower visual field are rather close to Gaussian distributions and that extreme non-Gaussian distributions occur near the horizontal meridian.

A comparison with the direction distributions in the non-stabilized condition (second column) and the mixed depth condition (third column) shows the influence of the statistics of the ego-motion parameters and the depth on the distributions of retinal directions. Whereas the cloverleaf structure in the mean exists in all conditions, the patterns of scatter, skewness, kurtosis and negentropy show clear differences between the conditions. In the non-stabilized and mixed depth conditions, there are no differences between upper and lower visual field. Kurtosis and negentropy do not take as high values as some domains of the visual field in the

natural condition. Although the distributions in both conditions are similar skewed, natural motion parameters in combination with natural scenes have a cumulative effect on skewing for some regions of the visual field.

### 3.2.2 Properties of the distributions of retinal speed

The Figure 8 shows the visual field maps for mean, scatter, skewness, kurtosis, and negentropy for the distributions of retinal speed for the three conditions. Figure 8 A1 shows the mean of retinal speed for all position in the field of view in the natural condition. Mean retinal speed and scatter is zero at the center of view because of the assumed gaze stabilization. Mean retinal speed increases in the periphery up to 20 degrees per second. Note that the absolute values of speed would scale with walking speed, which was a constant 1.4 meter per second in our calculations, but only by a constant factor for all flow speeds. Hence, walking speed does not change the distribution over the field of view.

The increase of the mean speed is larger for the lower visual field than for the upper visual field. The scatter, ranging from 0 to 11 degrees per second, on the other hand, increases more in the upper visual field than in lower visual field. These differences between mean and scatter show that the retinal speeds are faster and more uniform in the lower visual field and slower and more variable in the upper visual field.

Since the appearance of the estimated distributions for speed suggests that these distributions are Gaussians on a logarithmic scale, we measured the skewness, kurtosis and negentropy for the logarithms of speed. These measures show how close the speed distributions are to logarithmic Gaussian distributions. Figure 8, Panels A3, A4 and A5 show the estimated values for the log-skewness, log-kurtosis and log-negentropy. These values are largely uniform over the visual field. The log-skewness ranges from  $-0.8$  to  $0.8$ , the log-kurtosis from  $0.5$  to  $3$ , and the estimated log-negentropy from  $0.03$  and  $0.11$ . Although these values are rather low, for each position either the skewness, or the kurtosis, or both values are significantly different from zero, which we tested by the calculating the standard errors and using the resulting error bars (approx. two times the standard error) as significance criterion. Therefore, the distributions of retinal speed are significantly different from log-Gaussian distributions. However, the small values of log-skew, log-kurtosis and log-negentropy may for practical purposes allow to model the distributions of retinal speed by log-Gaussian distributions.

The distributions in the non-stabilized condition (see Figure 8 column B) look very similar to those for the natural condition except for positions close to the center of view. Close to the center of view the mean and scatter of retinal speed do not vanish in the non-stabilized condition and do not fall below 1.3 degree per second and 1.6 degree per second, respectively. In the mixed depth condition, the distributions show a complete different pattern (see Figure 8 column C). The increases of the mean and scatter towards the periphery are not as pronounced as in the other conditions and do not rise about 8 degree per second and 7 degree per second, respectively. Log-skewness takes only negative values across the visual



field. Log-kurtosis is smaller than in the other conditions. Log-negentropy is in the same range as in the natural and non-stabilized condition. These results suggest that the depth statistics has a shaping effect on the statistics of retinal speed while the gaze stabilization reflex affects the statistics of retinal speed only in the center of the visual field.

### 3.3 Dependence of the local optic flow statistics on scene structure and ego-motion

In this section we describe the statistical dependencies of the retinal flow on the depth statistics of the scene (depth  $Z$  and fixation depth  $Z_f$ ) and on the heading direction ( $H_\phi, H_\theta$ ). The dependence on scene statistics is interesting because the speed of an element of the optic flow depends on the distance of the element from the observer. Moreover, in case of combined observer translation and gaze stabilization the speed and the direction of the motion vector of the optic flow element depends on the relationship between the distance of the element from the observer and the depth of the gaze point. Without gaze stabilization, the statistics of the retinal direction solely depend on the statistics of the heading direction and not on the statistics of depth. Retinal speed is only influenced by depth and not by the depth of the gaze point.

#### 3.3.1 Dependence of the local optic flow statistics on the depth statistics of the scene

To reveal to what extent the optic flow in the natural condition is statistically dependent on the depth statistics of the natural environment, we estimated the normed mutual information between the random variables retinal flow direction  $\Phi_{dir}$  and speed  $V$  and the following random variables of the scene structure: the two-dimensional vector  $(1/Z, 1/Z_f)$ , the inverse depth-values  $\frac{1}{Z}$ , and the quotient between the depth and the depth of the fixation point  $\frac{Z}{Z_f}$ . We take  $Z/Z_f$  rather than  $1/Z_f$  as a single random variable because the direction of retinal motion depends on  $Z/Z_f$  not on  $Z_f$  alone. Figure 9 column A shows the distribution of the normed mutual information over the visual field for the above parameter combinations.

The estimated values for  $mI^{normed}(\Phi_{dir}, (1/Z, 1/Z_f))$  (Figure 9 A1) range from 0.05 to 0.6. Positions in the lower visual field show smaller values than positions in the upper visual field. The highest values are observed along the horizontal meridian. The estimated values for  $mI^{normed}(\Phi_{dir}, 1/Z)$  (Figure 9 A2) are very small and nowhere exceed 0.16. The estimated values for  $mI^{normed}(\Phi_{dir}, Z/Z_f)$  (Figure 9 A3) show a similar distribution as  $mI^{normed}(\Phi_{dir}, (1/Z, 1/Z_f))$ , but with a peak of higher dependence (0.8) in the center of view. Taken together, these plots show that the dependence of the direction of the optic flow on the depth statistics of the scene is particularly strong in the upper visual field and that the combination of depth  $Z$  and fixation depth  $Z_f$  considerably influences the flow directions in this area. In contrast, the flow directions in the lower visual field do not carry

much information about scene structure. Finally, the statistics of  $1/Z$  by itself has hardly any influence on the statistics of the retinal direction (Figure 9 A2).

In the mixed-depth condition (Figure 9 column B), the dependence of direction on depth shows only minor variation over the visual field. The differences between upper and lower visual field in the natural conditions disappear in the mixed-depth condition. The statistical dependence of depth ( $1/Z$ ) on the statistics of flow directions remains minor. The non-stabilized condition is not shown, because in this condition retinal direction does not depend on the depth structure of the scene, and all values would be zero.

Figure 10 shows the statistical dependence of retinal speed on the depth structure of the scene. Compared to retinal direction (Figure 9 A), retinal speed shows an almost opposite dependence on the statistics of depth in the scene in the natural condition (Figure 10 column A). The estimated values for  $mI^{normed}(V, (1/Z, 1/Z_f))$  vary between 0.6 and 0.97 (Figure 10 A1), with smaller values in the lower visual field and high values in the upper visual field. The estimated values for  $mI^{normed}(V, 1/Z)$  (Figure 10 A2), range from 0.58 to 0.97 and show a visual field distribution similar to that of  $mI^{normed}(V, (1/Z, 1/Z_f))$ . The estimated values for  $mI^{normed}(V, Z/Z_f)$  (Figure 10 A3) range between 0.05 and 0.5. They show a minor influence of  $Z/Z_f$  on the statistics of retinal speed for the lower visual field and a moderately increased statistical dependence for the upper visual field. The latter is caused by an increased statistical dependence between  $1/Z$  and  $Z/Z_f$  for the upper visual field (data not shown). We therefore conclude that the dependence of the distributions of retinal speed on the depth statistics of the scene is mainly carried by  $1/Z$ .

In the non-stabilized condition, retinal speed depends only on  $1/Z$ . The estimated statistical dependence of retinal speed on  $1/Z$  resembles the estimated data in the natural condition (see Figure 10 B2 ). In the mixed-depth condition, retinal speed is highly dependent on depth at all positions of the visual field but the dependence on depth is symmetric between upper and lower field.

We find that considering scenes with a non-natural depth statistics results in completely modified statistical dependencies between depth and retinal optic flow. Thus, we conclude that these dependencies are specific in the case of natural scenes.

### 3.3.2 Dependence of the local optic flow statistics on the statistics of heading

The dependence of the statistics of the retinal direction on the statistics of the observer's heading is shown in Figure 11. Column A depicts the natural condition. The estimated values for  $mI^{normed}(\Phi_{dir}, (H_\phi, H_\theta))$  (Figure 11 A1), range between 0.5 and 0.99 and thus reveal a strong statistical influence of heading on the flow direction. This influence is much more pronounced in the lower visual field than in the upper visual field. This observation is in accordance with the observation in the previous section that the depth statistics have a larger influence in the upper visual field than in the lower visual field. This increased

dependence on depth in the upper visual field disturbs the linkage to the heading.

Panels A2 and A3 show how the retinal flow directions are influenced by the vertical and horizontal heading components  $H_\theta$  and  $H_\phi$  separately.  $H_\phi$  has a strong influence on the statistics of retinal flow direction for eccentric positions on the vertical meridian.  $H_\theta$  has the highest influence at eccentric positions along the horizontal meridian.

In the non-stabilized condition, retinal direction is completely predicted by the heading direction, as all retinal motion is radially away from the heading point (focus of expansion). This means that the mutual information between the distribution of directions and heading is infinite. However, as in the natural condition the different components of heading have different statistical influence on retinal direction for different domains of the visual field, but the generated bulges of high normed mutual information values are broader than in the natural condition and do not show the abrupt decrease in the center of the visual field (see Figure 11, Panels B2 and B3). According to the certainty that in the non-stabilized condition the depth structure of the scene has no influence on the statistical behavior of flow directions, there are no differences of the estimated mutual information values between the upper and the lower visual field. In the mixed depth condition the statistical influence of heading on the statistics of flow directions are decreased for the lower visual field and increased for the upper visual field compared with the natural condition (see Figure 11 column C). This is a result of the influence of the parameter  $Z/Z_f$  on the statistics of retinal flow direction (see Figure 10 B3). The distribution of normed mutual information for the components  $H_\phi$  and  $H_\theta$  is similar to that of the natural condition, but rather than drop near the center of view as in the natural condition, the distributions rise in the center of view (see Figure 11, Panels C2 and C3).

Figure 12 shows the dependence of the statistics of the retinal speed on the statistics of the observer's heading. In the natural condition (Figure 12 A1), the estimated values for  $mI^{normed}(V, (H_\phi, H_\theta))$  (Figure 12 A1), which vary between 0 and 0.77, indicate only a modest statistical influence of heading on the statistics of retinal speed. Similar to retinal direction, the influence is more pronounced in the lower visual field than in the upper visual field.

The restriction of the statistical influence to the lower visual field is also seen for the separate vertical and horizontal heading components  $H_\theta$  and  $H_\phi$  (Figure 12 A2 and A3). This correlates with the diminished effect of the statistics of depth on the retinal speed for the lower visual field. However, the heading components  $H_\phi$  and  $H_\theta$  assert their influence in different parts of the lower visual field.  $H_\phi$  influences retinal speed along the diagonals in the lower visual field whereas  $H_\theta$  influences retinal speed near the vertical meridian in the lower visual field.

In the non-stabilized condition (Figure 12, column B) there is more statistical dependence in the upper visual field but the dependence in the lower visual field is very similar to that in the natural condition. In the mixed depth condition, heading has hardly any statistical influence on retinal speed (Figure 12 column C).

Together, the different influences of  $H_\phi$  and  $H_\theta$  on retinal direction (Figure 11) and retinal speed (Figure 12) can be explained by the following consideration: A horizontal deviation of heading from straight ahead causes a deviation of the retinal flow direction from radial for the flow elements close to the vertical meridian. Close to the horizontal meridian the direction of flow vectors is less affected, because mainly the speeds are increased or reduced there. Analogously, the statistical dependence of flow direction along the horizontal meridian is higher for the vertical heading direction. However, the counter-rotation of the retina in the case of gaze stabilization affect the statistical influence of the components of heading by keeping the directions closer to radial.

These observations show that the statistical dependencies of retinal optic flow on heading are shaped by both the geometry of natural scenes and the properties of natural ego-motion. Altering either the ego-motion parameters, or the depth statistics, considerably changes the dependence structure between retinal optic flow and heading.

### 3.4 Summary and Discussion

The results show how the structure of the retinal flow depends on the scene statistics and the ego-motion statistics. The principle dependence of the retinal flow on these parameters is clear from the geometrical properties of flow generation (Longuet-Higgins & Prazdny, 1980). However, the particular relevance of individual parameters in natural situations depends on the statistics of these parameters in the natural context. In the natural condition, the random variables retinal speed and retinal direction show rather low statistical interdependencies at almost all positions in the visual field. This statistical independence between retinal speed and retinal direction in the natural condition allows to efficiently encode both parameters independently, as is the case in motion sensitive neurons in visual cortical area MT. These neurons have largely independent tuning curves for direction and speed (Rodman & Albright, 1987). Gaze stabilization plays an important role for the independence between retinal speed and direction. Without gaze stabilization there is a much higher statistical interdependence between retinal speed and retinal direction for large domains of the lower visual field. Since, this increase occurs only in the lower visual field the depth structure appears to have also a strong influence on these interdependencies.

The distributions of retinal speed and retinal direction are strongly influenced by the underlying statistics of depth and ego-motion parameters. The statistical properties of the distributions measured for the different conditions differ strongly in their behavior over the field of view. In the natural condition, differences between the upper and the lower visual field are clearly visible for both retinal flow direction and retinal speed. In the non-stabilized condition, in contrast, differences between upper and lower visual field only occur in the distributions of retinal speed. This is because depth has no influence on flow direction in the non-stabilized condition. The statistical differences for the upper and the lower visual field are caused by different depth statistics for the upper and the lower visual field. Most natural scenes consist of objects on a ground surface. The ground surface may be flat, or form dips

and humps, or it can decline or rise. But in each scene, the ground confines the maximal depth at each positions of the visual field. Therefore, the existence of a ground in natural scenes generates an asymmetry in the depth statistics between positions in the upper and the lower visual field and restricts the variability of depth in the lower visual field. This asymmetry underlies all observed asymmetries between upper and lower visual field in the flow statistics. When the asymmetry in the depths statistics are destroyed in the mixed-depth condition, the differences in optic flow statistics between the upper and the lower visual field vanish. All statistical variations between different positions in the field of view in that condition are caused by the mathematical rules behind optic flow generation and the statistics of heading. However, the asymmetries between upper and lower visual field in the natural condition arise not just from the depth distribution alone, but rather from a combination of the depth distribution and the natural ego-motion parameters. For instance, many properties of the flow distributions in Figure 11 are symmetric between upper and lower field also in the non-stabilized gaze condition. Regarding the properties of early motion detectors our results coincides with the findings in Zanker & Zeil (2005), who also state differences in the distributions of the responses of early motion detectors between the upper and lower visual field for straight motion through natural scenes.

One may predict properties of motion sensitive neurons from the statistics of retinal optic flow according to the principle of efficient encoding, particular with respect to the dependence of tuning properties on the positions of the receptive field. The variation of the distributions of retinal speed and directions over the visual field may explain the variation of properties of neurons encoding different positions in the visual field. Thus, populations of neurons encoding for optic flow near the center of the visual field should be more sensitive for low speed but for a large range of retinal directions, whereas populations of neurons encoding for optic flow more peripherally should be more sensitive for large retinal speed but for more radial retinal directions (cf for instance (Albright, 1989) for such distributions in the primate cortical area MT). The tuning curves of such neurons should account for quantities such as skew and kurtosis, which might effect the proportion of sharply and broadly tuned neurons as well as the tuning in individual cells. Furthermore, the peripheral increase of the size of the receptive field of motion processing neurons in area MT seems to be well adapted to the structure of natural flow fields (Calow et al., 2005). More quantitative predictions may be derived from an analysis of efficient encoding of optic flow based on the measured distributions.

Our analysis reveals that the dependence of retinal speed and direction on the set of ego-motion and scene parameters varies considerably across the visual field. In the natural condition, the influence of the depth statistics on the retinal speed is strongest in the upper visual field and much weaker in the lower visual field. Since optic flow depends on depth and heading, an increase in the statistical influence of one parameter must be accompanied by a decrease of the statistical influence of another parameter. Therefore, the reduction of the influence of the depth statistics on the retinal speed coincides with an increased influence of the statistics of heading on the statistics of retinal speed at the lower visual field. This is true also in the non-stabilized condition. The finding that the dependence of retinal speed on depth is highest in the mixed depth condition shows that retinal speed can be regarded as

directly linked to the depth map of the scene, at least for the upper visual field. For the lower visual field, the dominance of the ground and the associated decrease in the variation of the depth over different scenes increases the statistical influence of the remaining parameters.

The statistics of retinal direction in the natural condition, on the other hand, are independent of the statistics of depth throughout most of the visual field. The same is true in the mixed depth condition. However, retinal direction is linked to the statistics of the combination of depth and depth of fixation in terms of the quotient  $Z/Z_f$ . This quotient separates the scene in foreground (entities closer than the fixation point,  $Z/Z_f < 1$ ) and background (entities more distanced than the fixation point,  $Z/Z_f > 1$ ). The dependence of direction on  $Z/Z_f$  is most pronounced near the horizontal meridian, presumably because variation in depth relative to the depth of fixation occurs most frequently in that area of the scene image.

It is conceivable that the tight statistical linking of optic flow to the depth structure of the scene enables the brain to reconstruct a good relative depth map of the scene from the motion signals. In the natural condition, heading influences the statistics of direction and speed of the retinal motion to different degrees. This is especially pronounced in the lower visual field. Heading has the largest influence on the statistics of direction. The influence of heading on retinal speed remains minor. The azimuth and the elevation component of heading have mutually exclusive statistical influence on the retinal flow direction. The azimuth component and the elevation component of heading are statistically independent ( $mI^{normed}(H_\phi, H_\theta) = 0.0032$ ) in the distribution we used. The strong statistical influence of one heading component in a certain domain of the visual field leads to a high correlation between the directions of flow vectors within that domain and to a lower correlation between the directions of flow vectors lying in domains which are influenced by the other heading component. We leave the quantitative investigation of the statistical correlation between flow vectors at different position of the field of view and the extraction of possible independent components or patterns for future work. But detection of such flow patterns requires receptive fields, which fully contain the extend of the pattern. Therefore, the sizes of the domains of a high statistical influence of a certain heading component on retinal flow might also predict the sizes of the receptive fields of heading sensitive neuron. Heading estimation from optic flow is processed in the medial superior temporal (MST) brain area, which receives most of the incoming information from motion sensitive neurons in area MT and which is widely accepted to process patterns of optic flow (Duffy & Wurtz, 1991; Tanaka & Saito, 1989; Lappe, 1996). The large receptive field of neurons in area MST are therefore consistent with the large extend of the domains of a high statistical influence of the azimuth and the elevation component of heading on retinal optic flow, particularly in the periphery.

Our study was performed with human-like ego-motion. It is difficult to speculate how the local statistics of retinal velocity will differ for other animals. The height of the eyes above ground may quantitatively alter the statistics, particularly in the lower visual field. However, Zanker & Zeil (2005) reported that the properties of motion signal distributions in the upper visual do not change much with variation of the height of the camera field. Many higher animals that live mostly on the ground, in similar environments, and perform gaze-stabilization reflexes will qualitatively encounter similar local statistics of optical flow. To

what extend the statistics will change quantitatively for different species is an interesting question for future work.

**Acknowledgements** M.L. is supported by the German Science Foundation DFG LA-952/2 and LA-952/3, the German Federal Ministry of Education and Research, and the EC Project Drivisco.

## References

- Albright, T. D. (1989). Centrifugal directionality bias in the middle temporal visual area (MT) of the macaque. *Vis.Neurosci.*, *2*, 177–188.
- Atick, J. J. & Redlich, A. N. (1992). What does the retina know about natural scenes? *Neural Comp.*, *4*, 196–210.
- Barlow, H. B. (1961). Possible principles underlying the transformation of sensory messages. In Rosenblith, W. A., Ed., *Sensory Communication*, pages 217–234. MIT Press, Cambridge, MA.
- Berkes, P. & Wiskott, L. (2002). Slow feature analysis yields a rich repertoire of complex cell properties. *Proc. Int. Conf on Artificial Neural Networks, ICANN02*, pages 81– 86.
- Betsch, B.; W.Einhäuser; K.Koerding, & König, P. (2004). The world from a cats perspective - statistics of natural videos. *Biol.Cybern.*, *90*, 41– 50.
- Calow, D.; Krüger, N.; Wörgötter, F., & Lappe, M. (2004). Statistics of optic flow for self-motion through natural sceneries. In U.Ilg; Bühlhoff, H. H., & Mallot, H. A., Eds., *Dynamic Perception 2004*, pages 133–138.
- Calow, D.; Krüger, N.; Wörgötter, F., & Lappe, M. (2005). Biologically motivated space-variant filtering for robust optic flow processing. *Network: Computation in Neural Systems*, *16*(4), 323–340.
- Duffy, C. J. & Wurtz, R. H. (1991). Sensitivity of MST neurons to optic flow stimuli. I. A continuum of response selectivity to large-field stimuli. *J.Neurophysiol.*, *65*, 1329–1345.
- Elder, E. H. & Goldberg, R. G. (2002). Ecological statistics of Gestalt laws for the perceptual organization of contours. *J.Vision*, *2*, 324–353.
- Fermüller, C.; Shulman, D., & Aloimonos, Y. (2001). The statistics of optical flow. *Comp.Vis.Image Understand.*, *82*, 1–32.
- Gibson, J. J. (1950). *The Perception of the Visual World*. Houghton Mifflin, Boston.
- Gibson, J. J. (1966). *The Senses Considered As Perceptual Systems*. Houghton Mifflin, Boston.
- Huang, J.; Lee, A. B., & Mumford, D. (2000). Statistics of range images. *Proc. CVPR*, *1*, 324–321.
- Imai, T.; Moore, S. T.; Raphan, T., & Cohen, B. (2001). Interaction of the body, head, and eyes during walking and turning. *Experimental Brain Research*, *136*, 1–18.
- Ivins, J.; Porill, J.; Frisby, J. P., & Orban, G. (1999). The 'ecological' probability density function for linear optic flow: Implications for neurophysiology. *Perception*, *28*, 17–32.

- Kalkan, S.; Calow, D.; Felsberg, M.; Wörgötter, F.; Lappe, M., & Krüger, N. (2005). Local image structures and optic flow estimation. *Network: Computation in Neural Systems*, 16(4), 341–356.
- Kozachenko, L. F. & Leonenko, N. N. (1987). Sample estimate of the entropy of a random vector. *Probl. Peredachi Inf.*, 23, 95–101.
- Kraskov, A.; Stgbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Phys. Rev. E*, 69, 06613801–06613816.
- Krüger, N. (1998). Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2), 117–129.
- Krüger, N. & Wörgötter, F. (2002). Multi-modal estimation of collinearity and parallelism in natural image sequences. *Computation in Neural Systems*, 13, 553–576.
- Lappe, M. (1996). A model of a goal-directed integration of motion and stereopsis in visual cortex. *Perception*, 25, 133.
- Lappe, M. (2000a). Computational mechanisms for optic flow analysis in primate cortex. In Lappe, M., Ed., *Neuronal Processing of Optic Flow*, Int.Rev.Neurobiol.44, pages 235–268. Academic Press.
- Lappe, M., Ed. (2000b). *Neuronal Processing of Optic Flow*, Int.Rev.Neurobiol.44. Academic Press.
- Lappe, M.; Bremmer, F.; Pikel, M.; Thiele, A., & Hoffmann, K. (1996). Optic flow processing in monkey STS: a theoretical and experimental approach. *J. Neurosci.*, 16(19), 6265–6285.
- Lappe, M.; Bremmer, F., & van den Berg, A. V. (1999). Perception of self-motion from visual flow. *Trends.Cogn.Sci.*, 3, 329–336.
- Lappe, M.; Pikel, M., & Hoffmann, K.-P. (1997). Optokinetic eye movements elicited by radial optic flow in macaque monkeys. *Soc. Neurosci. Abstr.*, 23, 758.
- Lappe, M.; Pikel, M., & Hoffmann, K.-P. (1998). Optokinetic eye movements elicited by radial optic flow in the macaque monkey. *J.Neurophysiol.*, 79, 1461–1480.
- Laughlin, S. B. (1981). Simple coding procedure enhances a neuron’s information capacity. *Z.Naturforsch.*, 36C, 910–912.
- Longuet-Higgins, H. C. & Prazdny, K. (1980). The interpretation of a moving retinal image. *Proc.Royal.Soc.London B*, 208, 385–397.
- Niemann, T.; Lappe, M.; Bscher, A., & Hoffmann, K.-P. (1999). Ocular responses to radial optic flow and single accelerated targets in humans. *Vis. Res.*, 39, 1359–1371.
- Olshausen, B. A. & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381, 607–609.
- Parzen, E. (1962). On estimation of a probability density function and mode. *Ann. Math. Statistic*, 33, 1065–1076.
- Rodman, H. R. & Albright, T. D. (1987). Coding of visual stimulus velocity in area MT of the macaque. *Vis. Res.*, 27(12), 2035–2048.
- Roth, S. & Black, M. J. (2005). On the spatial statistics of optical flow. *IEEE Int. Conf. on Comp. Vision (ICCV)*, 1, 42–49.
- Rudermann, D. L. & Bialek, W. (1994). Statistics of natural images: scaling in the woods. *Phys.Rev.Let.*, 39, 814–817.



- Saito, H.-A.; Yukie, M.; Tanaka, K.; Hikosaka, K.; Fukada, Y., & Iwai, E. (1986). Integration of direction signals of image motion in the superior temporal sulcus of the macaque monkey. *J.Neurosci.*, *6*, 145–157.
- Silverman, B. W. (1986). Density estimation. *Chapman and Hall*.
- Simoncelli, E. P. & Olshausen, B. A. (2001). Natural image statistics and neural representation. *Ann.Rev.Neurosci.*, *24*, 1193–1216.
- Solomon, D. & Cohen, B. (1992). Stabilization of gaze during circular locomotion in light: I. compensatory head and eye nystagmus in the running monkey. *J.Neurophysiol.*, *67*, 1146–1157.
- Tanaka, K. & Saito, H.-A. (1989). Analysis of motion of the visual field by direction, expansion/contraction, and rotation cells clustered in the dorsal part of the medial superior temporal area of the macaque monkey. *J. Neurophysiol.*, *62*, 626–641.
- van Hateren, J. H. & Rudermann, D. L. (1998). Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proc.Royal.Soc.London B*, *265(1412)*, 2315–2320.
- Weiss, Y. & Fleet, D. (2001). Velocity likelihoods in biological and machine vision. In Rao, R. P. N.; Olshausen, B. A., & Lewicki, M. S., Eds., *Probabilistic Models of the Brain, Perception and Neural Function*, pages 81–100. The MIT Press, Massachusetts Institute of Technology.
- Zanker, J. & Zeil, J. (2005). Movement-induced motion signal distribution in outdoor scenes. *Network: Computation in Neural Systems*, *16(4)*, 357–376.
- Zetsche, C. & Krieger, G. (2001). Nonlinear mechanism and higher-order statistics in biological vision and electronic image processing: review and perspectives. *J.Electronic Imaging*, *10*, 56–99.

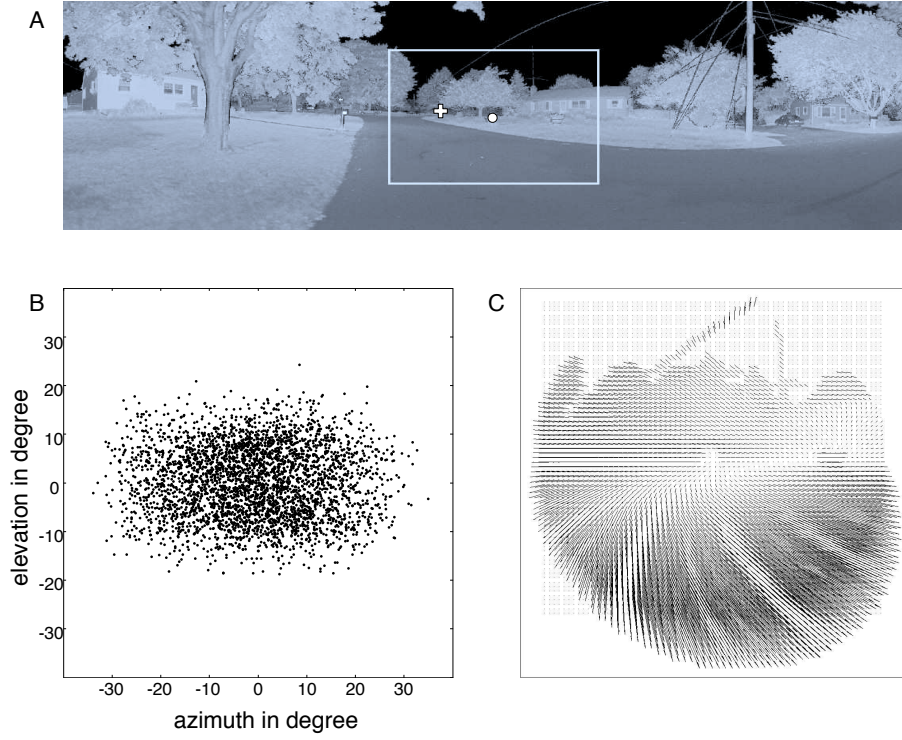


Figure 1: A: Panoramic projection of 3D data of a range-image, The grey values encode the intensity of the reflected laser beam. B: measured gaze directions projected onto the azimuth-elevation plane, C: Retinal flow field generated by a leftward motion and a gaze stabilizing eye movement through the scene depicted with the white frame in A. The motion direction is depicted by a cross. The direction of gaze is depicted by a disc.

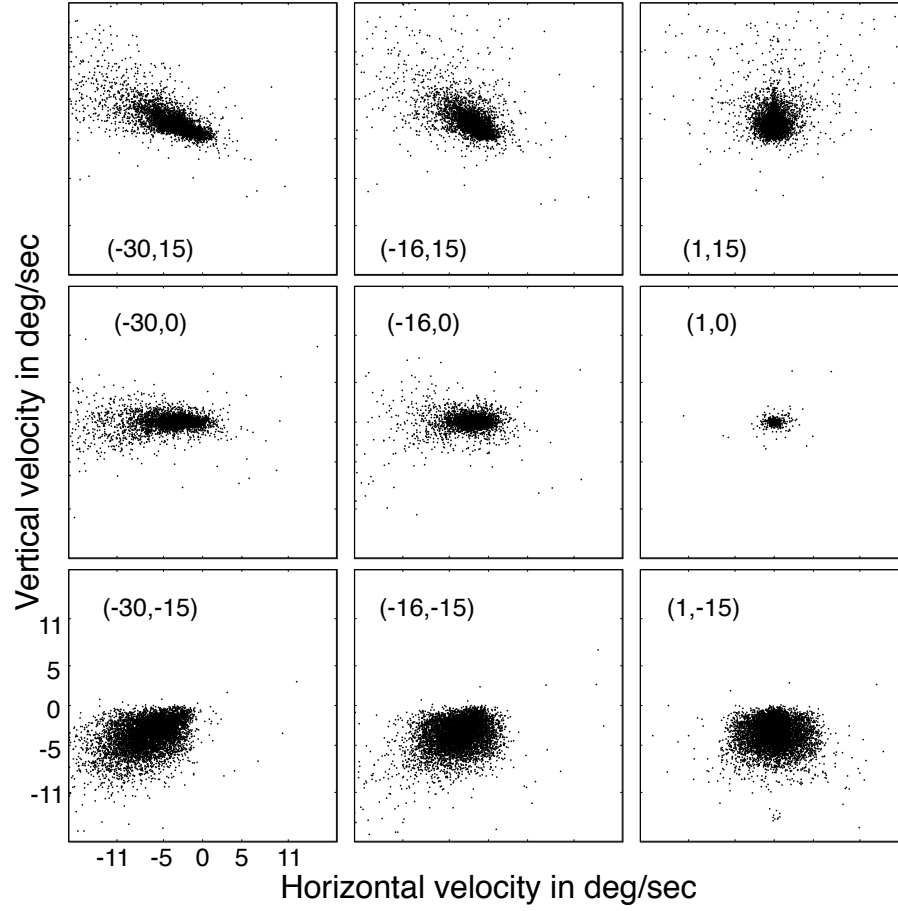


Figure 2: Measured distributions of retinal velocity for 9 different positions in the left visual field. The numbers in each panel give the visual field position in spherical coordinates  $(\tilde{\phi}, \tilde{\theta})$  in degrees. First row: positions in the upper field of view, middle row: horizontal meridian, third row: lower field of view.

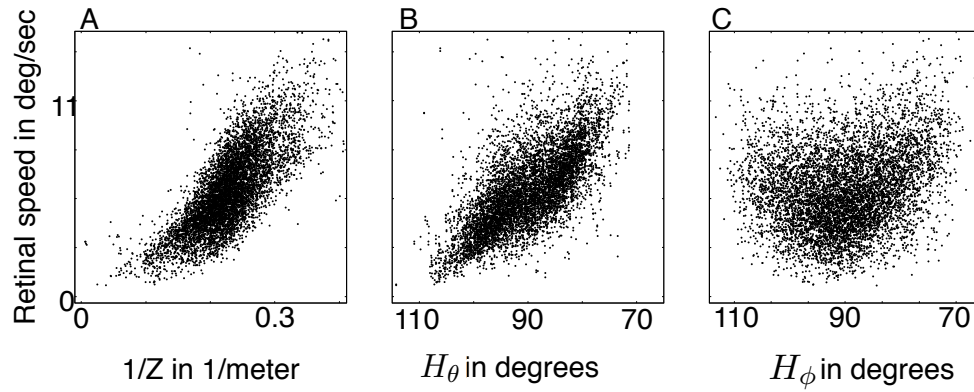


Figure 3: Examples of scatter plots of the dependence of retinal speed on inverse depth (A), elevation of heading (B), azimuth of heading (C). The data are from the visual field position  $(-5^\circ, -18^\circ)$ .

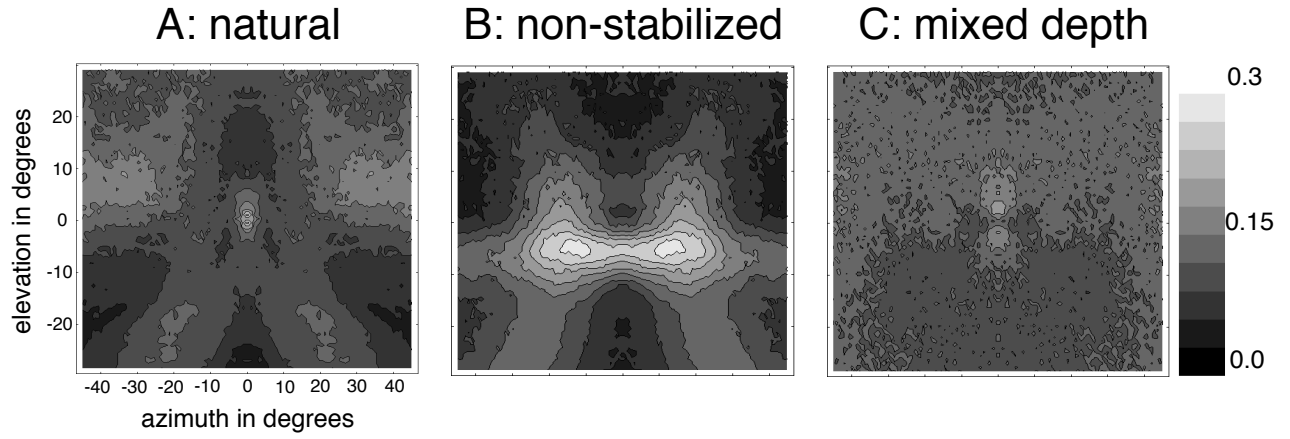


Figure 4: Statistical dependence between retinal speed and direction. Three-dimensional plots and contour plots of the estimated normed mutual informations  $mI^{normed}(\Phi_{dir}, V)$  between retinal flow direction and retinal speed as function of the position of the visual field for different conditions.

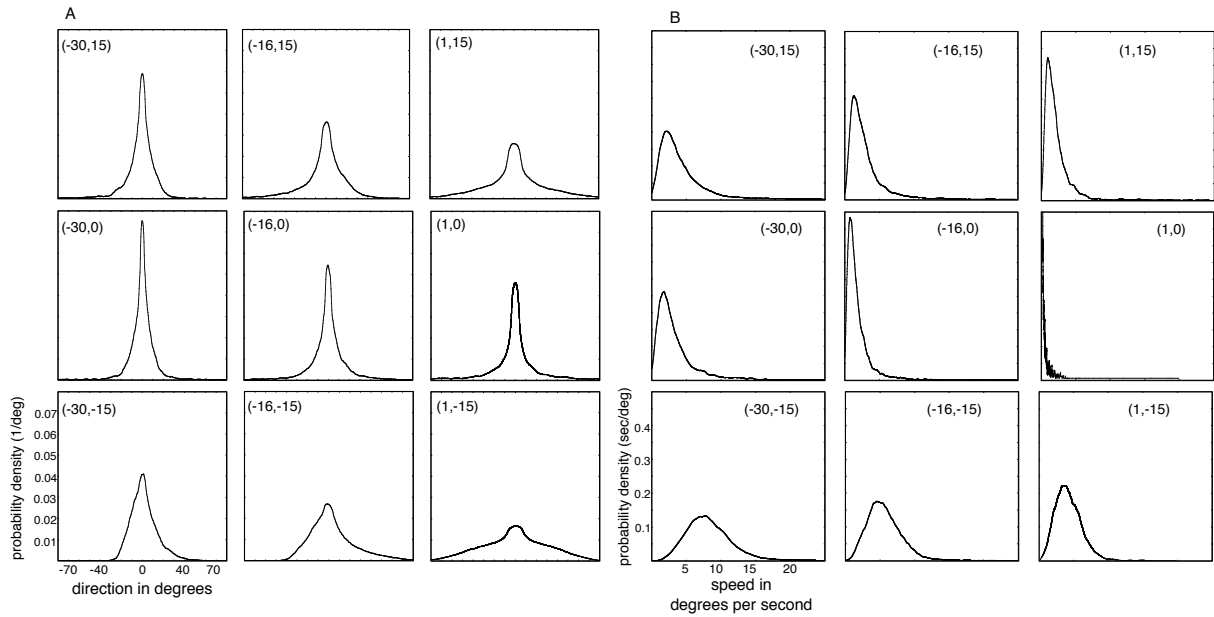


Figure 5: Examples of kernel-based density estimations for direction (A) and retinal speed (B). Nine positions in the left visual field are shown.

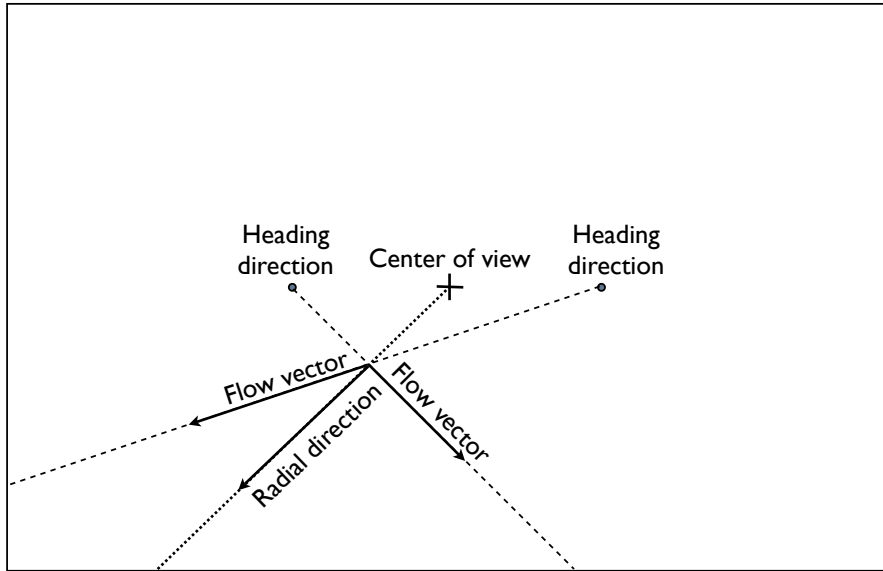


Figure 6: In the case of straight forward ego motion without rotations, the same horizontal deviation of heading to the left and to the right respectively results in different deviations from the radial direction for the resulting flow vectors. The distribution of flow directions is skewed.

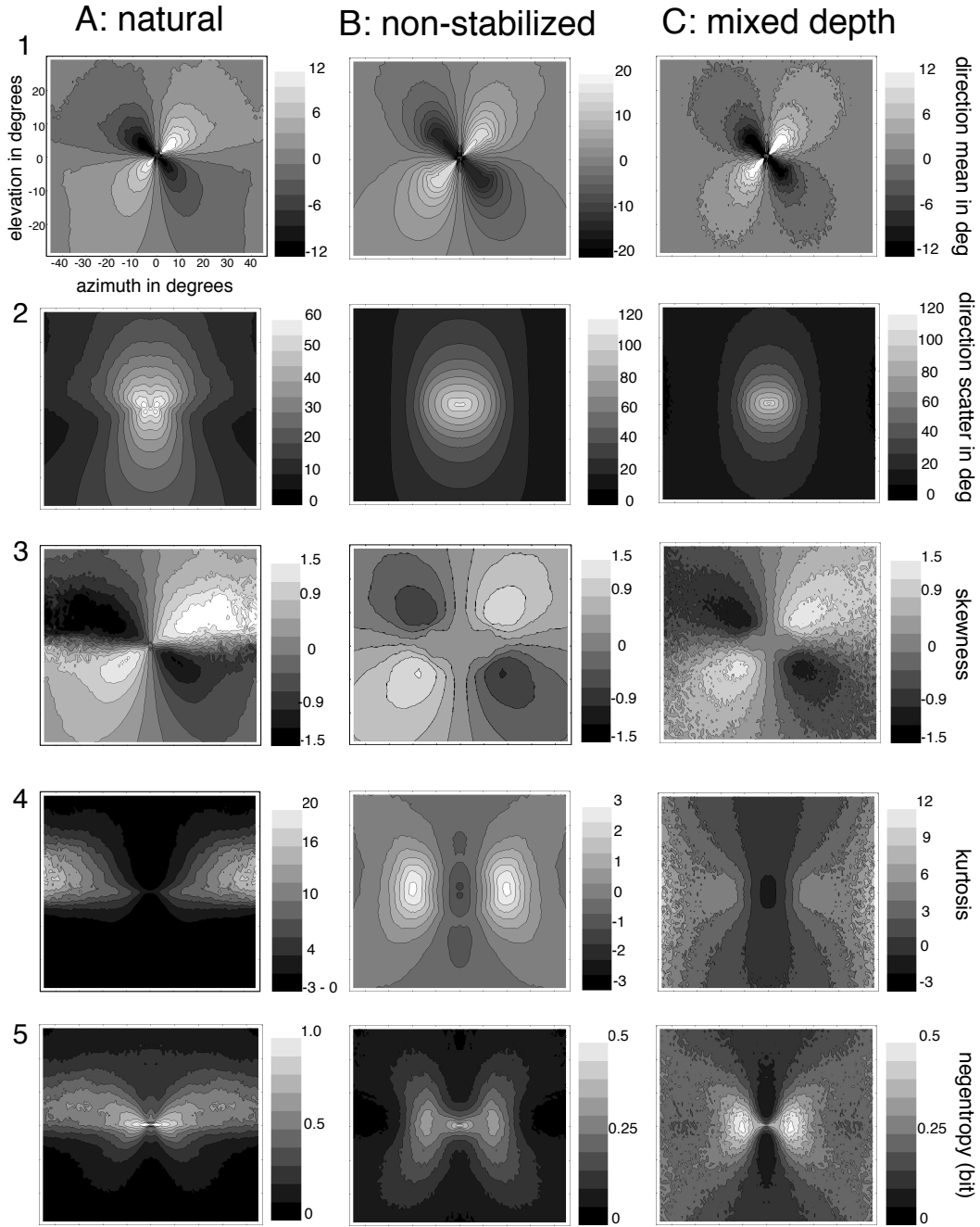


Figure 7: Estimated statistical properties of the distributions of directions for the different conditions plotted over the visual field. 1: estimated mean, 2: estimated scatter, 3: estimated skewness, 4: estimated kurtosis, 5: estimated negentropy

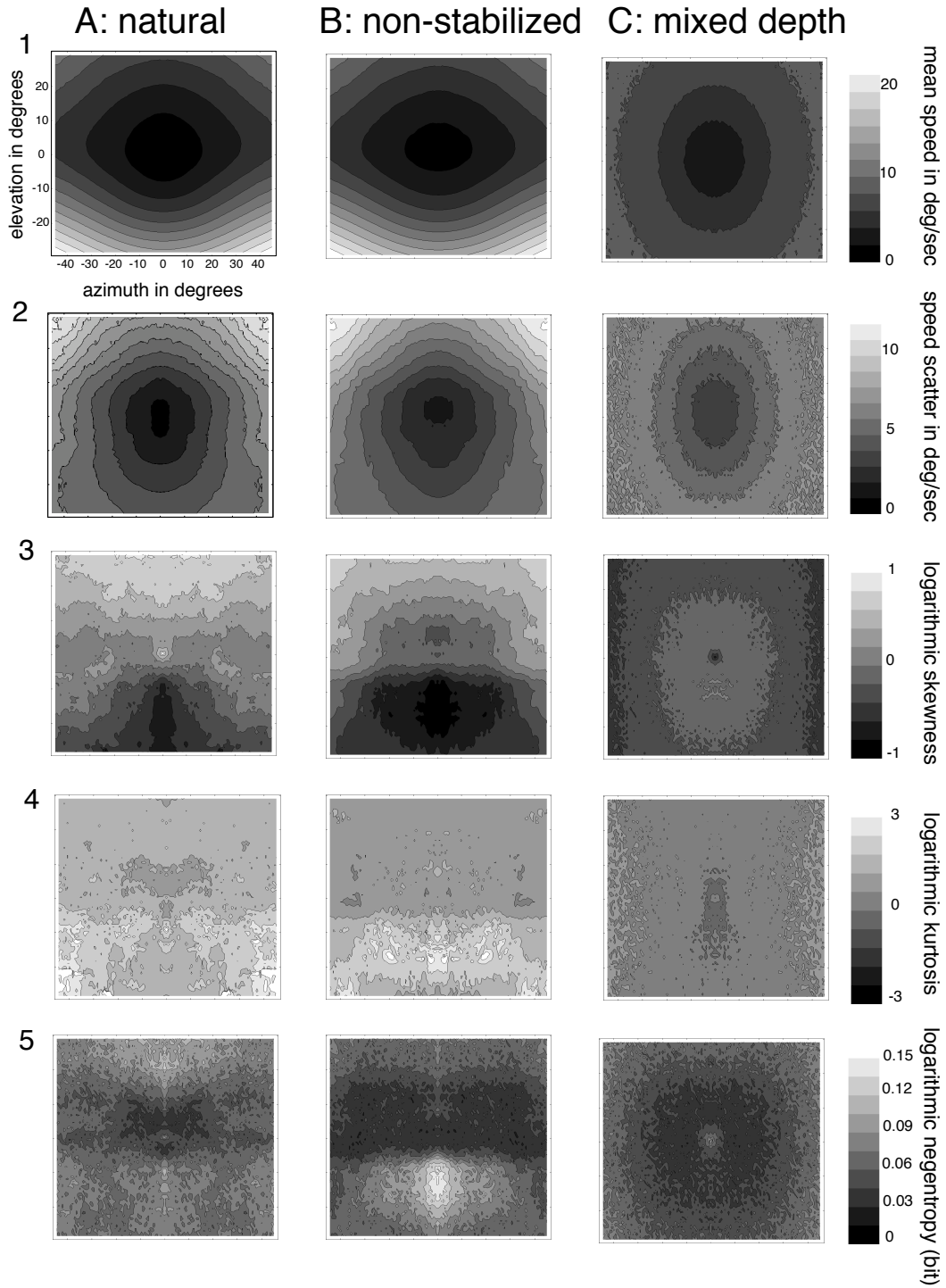


Figure 8: Estimated statistical properties of the distributions of retinal speed for the different conditions plotted over the visual field. 1: estimated mean, 2: estimated scatter, 3: estimated skewness of log-speed, 4: estimated kurtosis of log-speed, 5: estimated negentropy of the distributions of log-speed

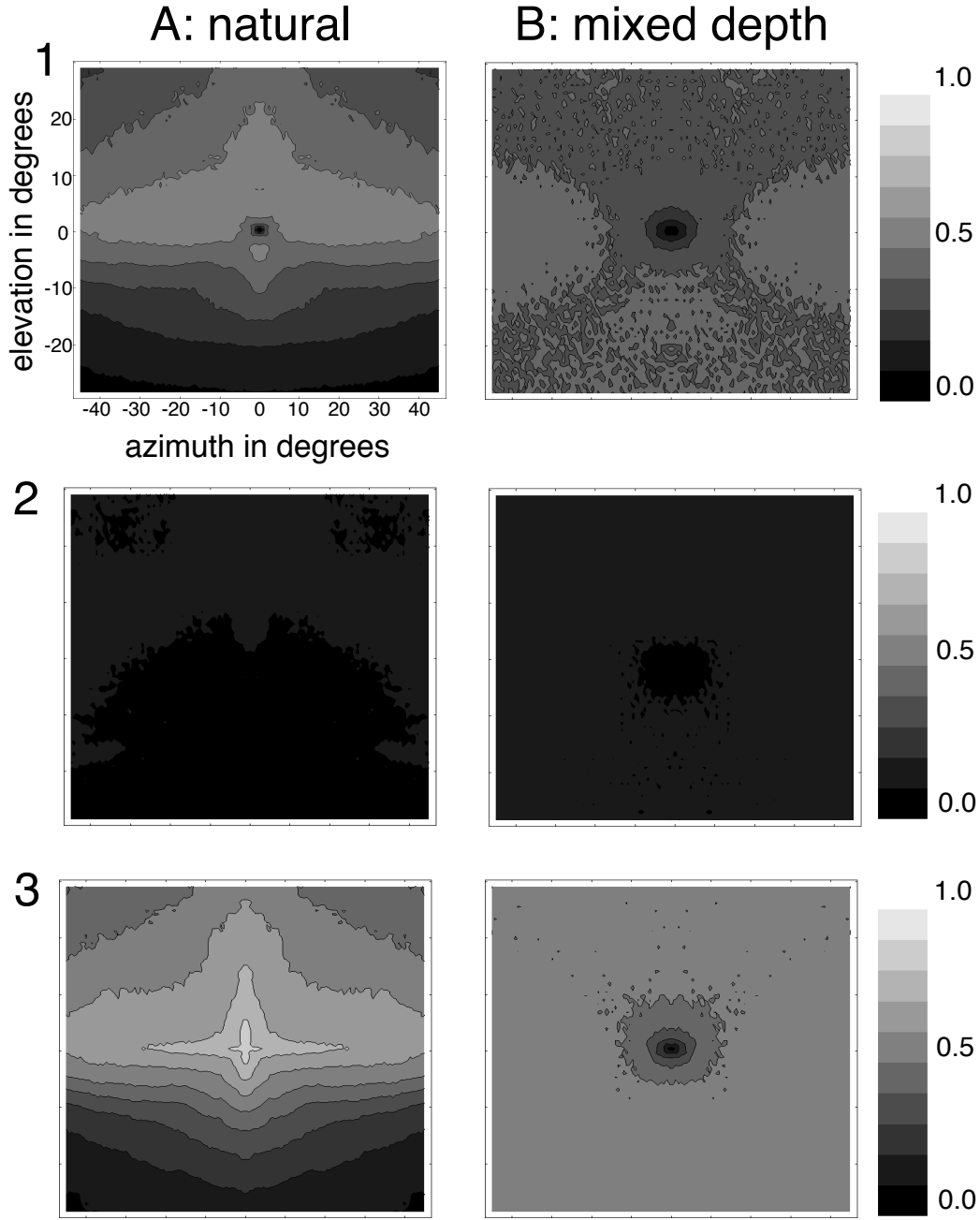


Figure 9: Statistical dependence between direction and depth. Estimated normed mutual information in the natural and the mixed depth condition plotted over the visual field. 1:  $mI^{normed}(\Phi_{dir}, (1/Z, 1/Z_f))$  between retinal direction  $\Phi_{dir}$  and depth structure  $(1/Z, 1/Z_f)$ , 2:  $mI^{normed}(\Phi_{dir}, 1/Z)$  between  $\Phi_{dir}$  and the inverse of depth  $1/Z$ , 3:  $mI^{normed}(\Phi_{dir}, Z/Z_f)$  between  $\Phi_{dir}$  and the quotient of depth and fixation depth  $Z/Z_f$ .



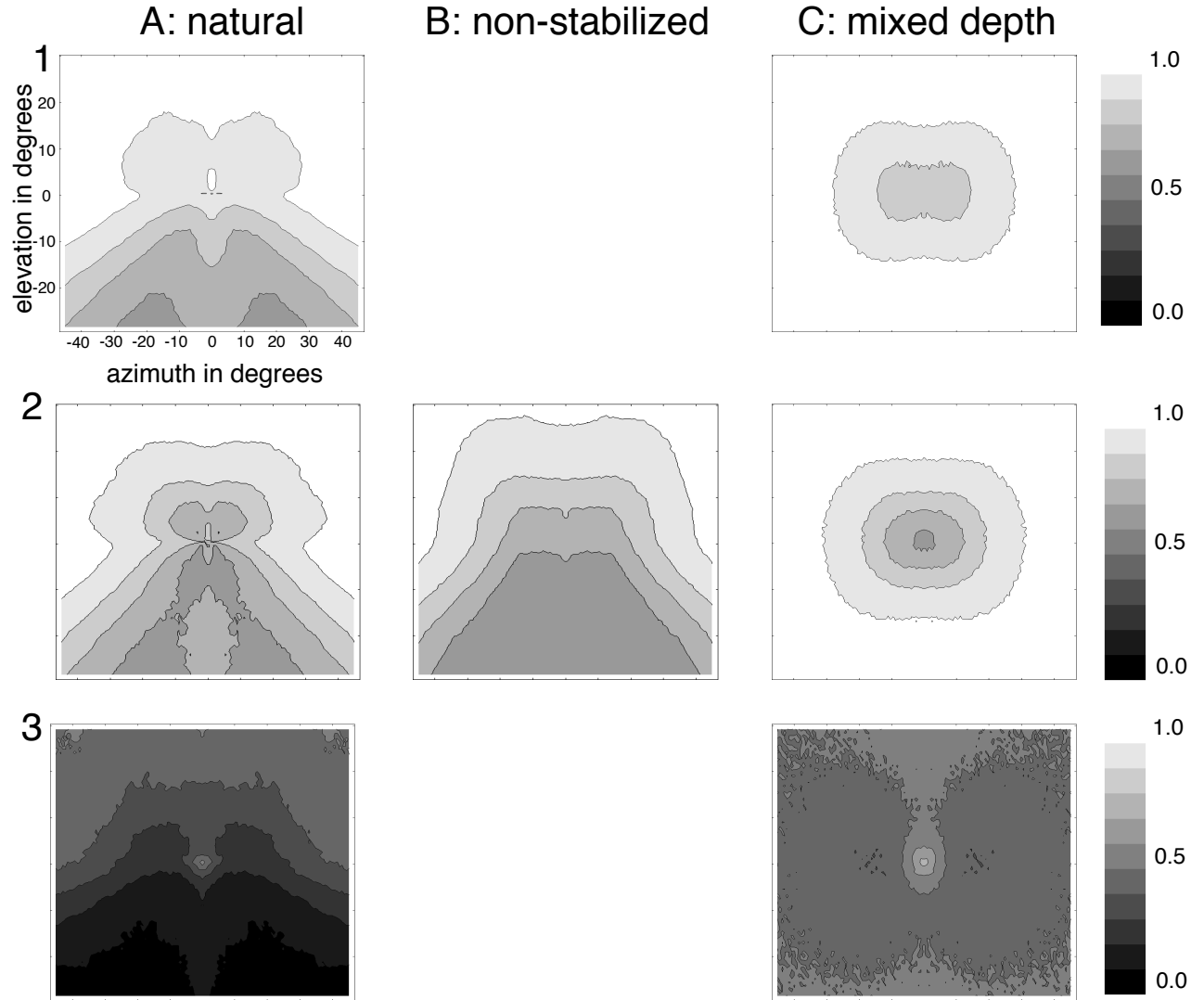


Figure 10: Statistical dependence between speed and depth. Estimated normed mutual information in the different conditions plotted over the visual field. 1:  $mI^{normed}(V, (1/Z, 1/Z_f))$  between retinal speed  $V$  and depth structure  $(1/Z, 1/Z_f)$ , 2:  $mI^{normed}(V, 1/Z)$  between  $V$  and the inverse of depth  $1/Z$ , 3:  $mI^{normed}(V, Z/Z_f)$  between  $V$  and the quotient of depth and fixation depth  $Z/Z_f$

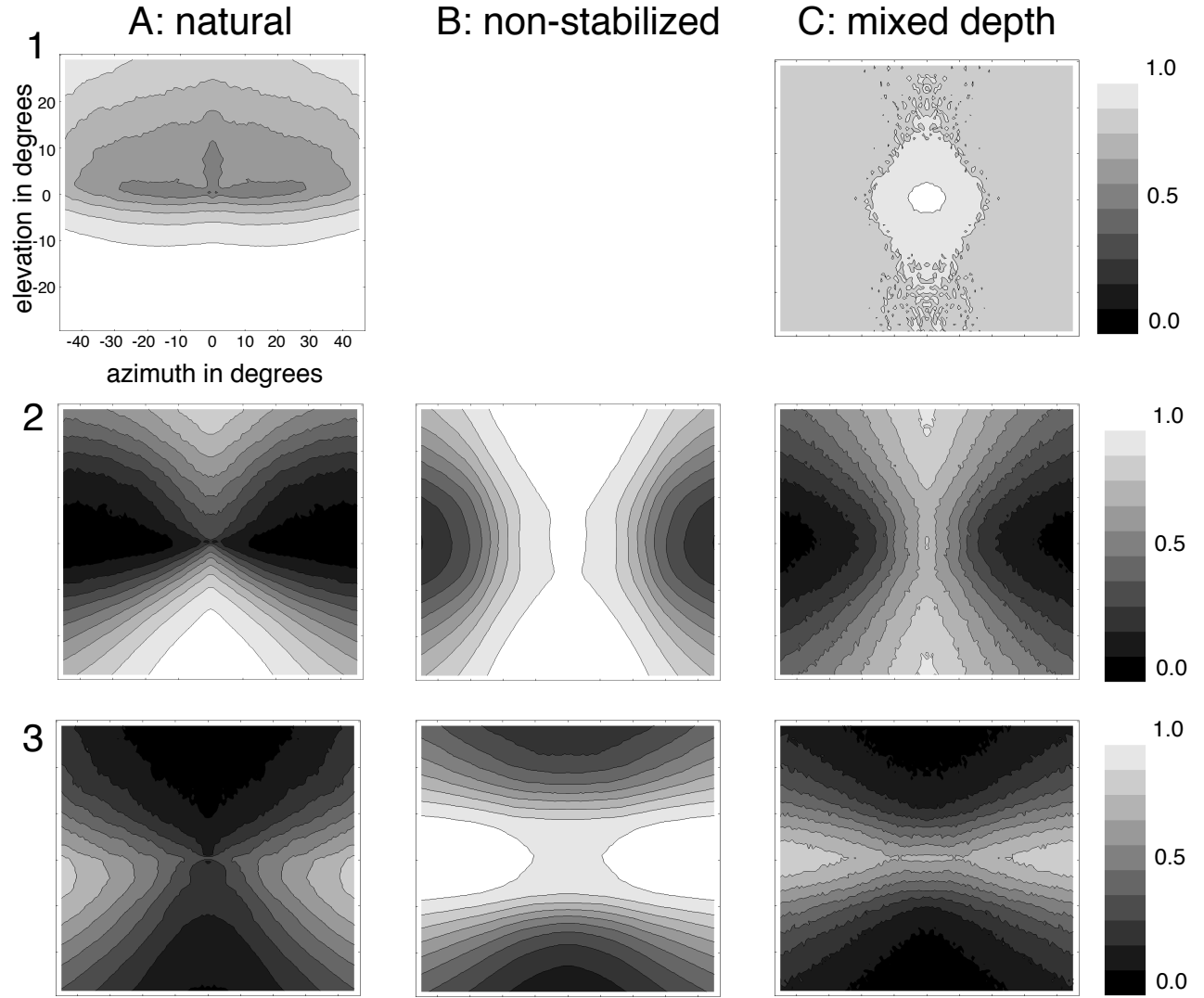


Figure 11: Statistical dependence between direction and heading. Estimated normed mutual information in the different conditions plotted over the visual field. 1:  $mI^{normed}(\Phi_{dir}, (H_\phi, H_\theta))$  between retinal direction  $\Phi_{dir}$  and heading  $(H_\phi, H_\theta)$ , 2:  $mI^{normed}(\Phi_{dir}, H_\phi)$  between  $\Phi_{dir}$  and the horizontal heading component  $H_\phi$ , 3:  $mI^{normed}(\Phi_{dir}, H_\theta)$  between  $\Phi_{dir}$  and the vertical heading component  $H_\theta$ .

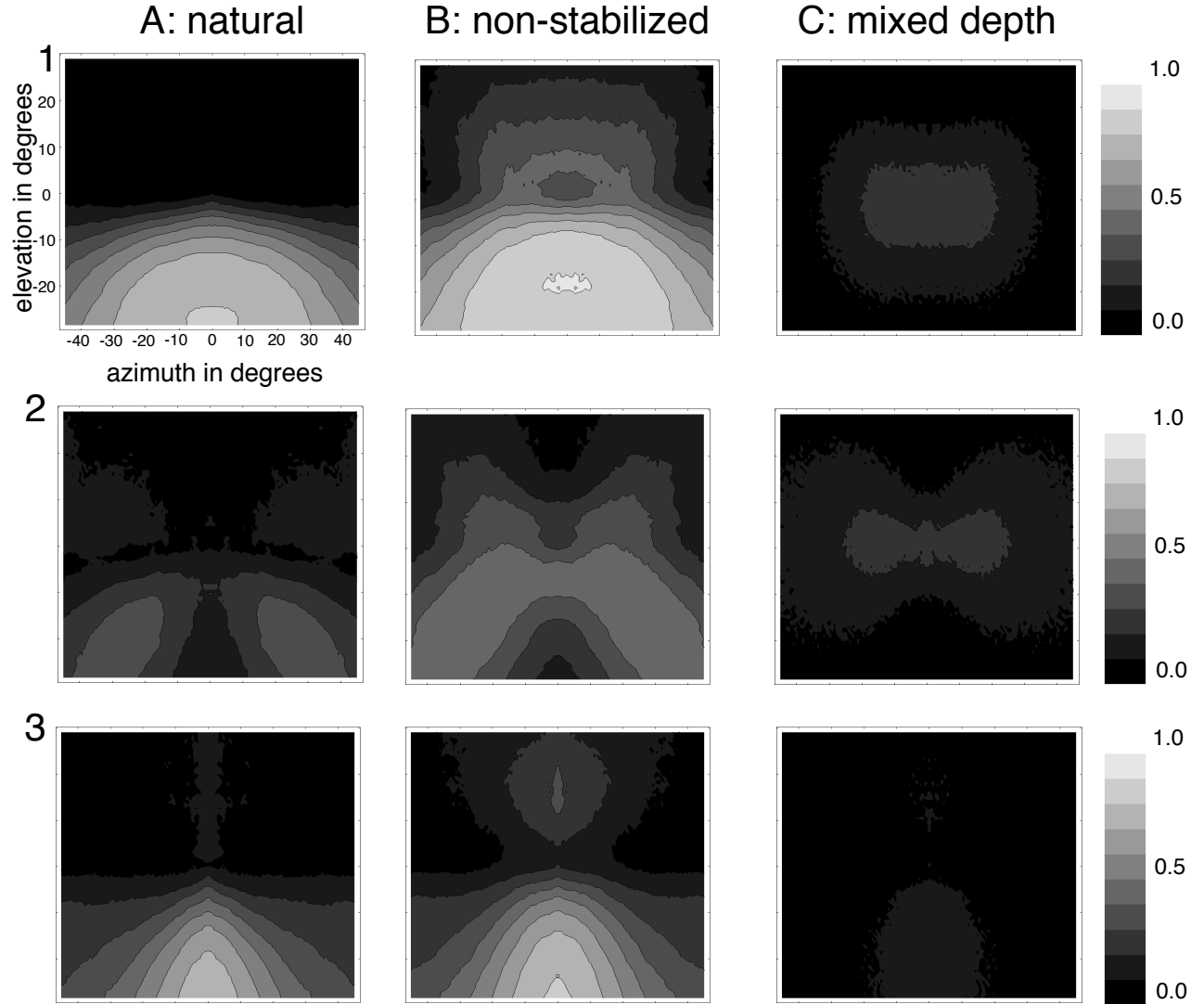


Figure 12: Statistical dependence between speed and heading. Estimated normed mutual information in the different conditions plotted over the visual field. 1:  $mI^{normed}(V, (H_\phi, H_\theta))$  between retinal speed  $V$  and heading  $(H_\phi, H_\theta)$ , 2:  $mI^{normed}(V, H_\phi)$  between  $V$  and the horizontal heading component  $H_\phi$ , 3:  $mI^{normed}(V, H_\theta)$  between  $V$  and the vertical heading component  $H_\theta$ .

## **Appendix C**

### ***Cue and Sensor Fusion for Independent Moving Objects Detection and Description in Driving Scenes***

N. Chumerin and M. Van Hulle

*( Signal Processing Techniques for Knowledge Extraction and Information Fusion – in press )*

---

# Cue and Sensor Fusion for Independent Moving Objects Detection and Description in Driving Scenes

Nikolay Chumerin and Marc M. Van Hulle

Katholieke Universiteit Leuven, Laboratorium voor Neuro- en Psychofysiologie,  
Campus Gasthuisberg, Herestraat 49, bus 1021, B-3000 Leuven, Belgium  
{nikolay.chumerin, marc.vanhulle}@med.kuleuven.be

## 1 Introduction

The detection of the *independently moving objects* (IMOs) can be considered as an exponent of the obstacle detection problem, which plays a crucial role in traffic-related computer vision. Vision alone is able to provide robust and reliable information for autonomous driving or guidance systems in real time but not for the full spectrum of real world scenarios. The problem is complicated by ego-motion, camera vibrations, imperfect calibrations, complex outdoor environments, insufficient camera resolutions and other limitations. The fusion of information obtained from multiply sensors can dramatically improve the detection performance [12, 31, 2, 3, 4, 9, 17, 30, 19, 13, 16, 5, 29, 18, 10, 32].

In Table 1 we present a chronological list of studies which are related to sensor fusion in traffic applications and which are relevant to the considered topic. Various sensors can be used for traffic applications: video (color or gray scale) cameras in different setups (monocular, binocular or trinocular), IR (infra red) cameras, LIDAR (Light Detection and Ranging), radar (Radio Detection and Ranging), GPS/DGP (Global Positioning System/Differential GPS) as well as data from vehicle IMU (Inertial Measurement Unit) sensors: accelerometer, speedometer, odometer and angular rate sensors (gyroscopes). There are a number of approaches to fusion characterization [11, 8, 26, 37] but, most frequently, fusion is characterized by the abstraction level:

1. Low (signal) level fusion combines raw data provided directly from sensors, without any preprocessing or transformation.
2. Intermediate (feature) level fusion aggregates features (e.g. edges, corners, texture) extracted from raw data before aggregation.
3. High (decision) level fusion aligns decisions proposed by different sources.

Depending on the application, several different techniques are used for fusion. Matching of the targets detected by different sensors is often used for

obstacle detection. Extensions of the Kalman filter (KF) [15] (e.g. extended Kalman filter (EKF) and unscented Kalman filter (UKF) [14]) are mostly involved in estimation and tracking of obstacle parameters, as well as in egoposition and egomotion estimation.

In this study, we propose a novel approach based on data fusion on different levels for IMO detection and -description. In the proposed model only three sensors are used: stereovision, speedometer and LIDAR. The flow diagram of the model is shown on Fig. 1. The IMOs detected by vision are matched with obstacles provided by LIDAR. In the case of a successful matching, the descriptions of the IMOs (distance, relative speed and acceleration) are retrieved using ACC (Adaptive Cruise Control) LIDAR data, or otherwise these descriptions are estimated based on vision. Absolute speed of the IMO is evaluated using its relative velocity and egospeed provided by the speedometer.

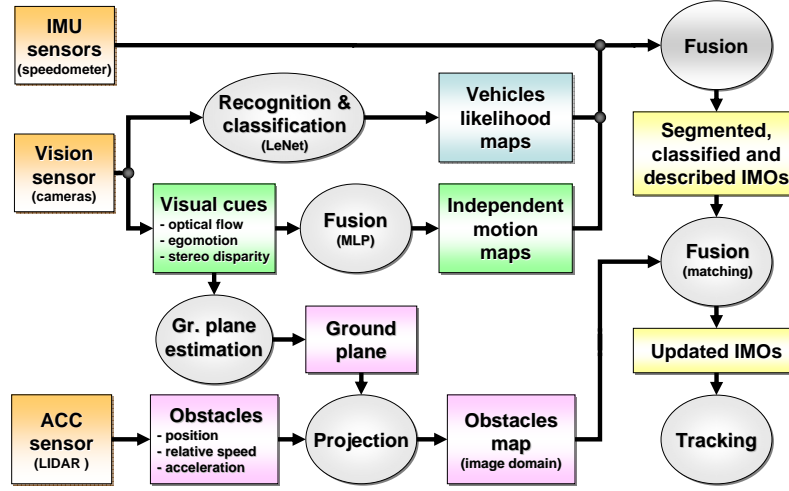


Fig. 1. Flow-diagram of the proposed model.

In order to validate the model we have used the data obtained in the frameworks of the DRIVSCO and ECOVISION European Projects. In recording sessions a modified Volkswagen Passat B5 was used as a test car. It was equipped by Hella KGaA Hueck & Co.

## 2 Vision sensor data processing

For vision-based IMO detection, we used an approach proposed by Chumerin and Van Hulle [7]. This method is based on the processing and subsequent fusing of two cooperative streams: the *independent motion detection stream*

**Table 1.** Sensor fusion for traffic applications papers short overview

Study	Sensors	Cues/Features	Fusion method
Handmann et al. [12]	monocular vision, radar	color, edges, texture (local image entropy), (up to 3) obstacle positions	MLP
Stiller et al. [31]	stereo vision, radar, LIDARs, DGPS/INS	horizontal edges, stereo disparity, optical flow, 2D range profile, global egoposition and egoorientation	Kalman filter
Becker and Simon [2]	stereo vision, DGPS, vehicle sensors, LIDARs, radar	local egoposition and ego-orientation (w.r.t. lane), global egoposition and ego-orientation, egospeed, egoacceleration, steering angle, 2D range profile	Kalman filter
Kato et al. [17]	video camera (monocular), radar	Kanade-Lucas-Tomasi feature points, range data	frame-to-frame feature points coupling based on range data
Fang et al. [9]	video cameras (stereo), radar	edges, stereo disparity, depth ranges	depth-based target edges selection and contour discrimination
Steux et al. [30]	color video camera (monocular), radar	shadow position, rear lights position, symmetry, color, 2D range profile	belief network
Hofmann et al. [13]	color video camera (monocular), BW video camera (monocular), radar, ACC-radarsensors	lane position and width, relative egoposition and ego-orientation (w.r.t. road), radar-based obstacles	extended Kalman filter
Laneurit et al. [19]	vision, GPS, odometer, wheel angle sensor, LIDAR	relative egoposition and ego-orientation (w.r.t. road), global egoposition and ego-orientation, steering angle, pathlength, LIDAR-based obstacle profile	Kalman filter
Sergi [28]	vision, LIDAR, DGPS	video stream, global egoposition and ego-orientation, LIDAR-based obstacle profile	Kalman filter
Sole et al. [29]	monocular camera, radar	horizontal and vertical edges, 'pole like' structures, radar target,	matchingpurpose
Blanc et al. [5]	IR camera, radar, LIDAR	IR images, range profile	Kalman filter, matching
Labayrade et al. [18]	stereo vision, LIDAR	stereo disparity, "v-disparity", lighting conditions, road geometry, obstacle positions	matching, Kalman filter, belief theory based association
Thrun et al. [33]	color video camera (monocular), GPS, LIDARs, radars, accelerometers, gyroscopes	color images, global egoposition and ego-orientation, egospeed, short-range profile (LIDARs), long-range obstacles (radars)	Unscdted Kalman Filter
Bombini et al. [6]	gray-scale video camera (monocular), radar	vertical edges symmetry, horizontal edges, radar-based obstacles	search and matching

and the *object recognition stream*. The recognition stream deals with static images (*i.e.*, does not use temporal information) and therefore can not distinguish between independently moving and static (*i.e.*, with respect to the environment) objects, but which can be detected by the independent motion stream. One should note that the idea of the two processing streams is widely accepted in the visual neurosciences [34].

## 2.1 Vision sensor setup

In the recording sessions, we used a setup with two high resolution progressive scan color CCD cameras (see Table 2). The camera rig was mounted inside the cabin of the test car (see Fig. 2) at 1.240 m height above the ground, with 1.83 m from the frontend and 17 cm displacement from the middle of the test car towards the driver’s side. Both cameras were oriented parallel to each other and to the longitudinal axis of the car and look straight ahead into the street. Before each recording session, the cameras were calibrated. Raw color (Bayer pattern) images and CAN-bus data were stored for further off-line processing. In the model, we used rectified gray-scale images downsampled to a  $320 \times 256$  pixels resolution.

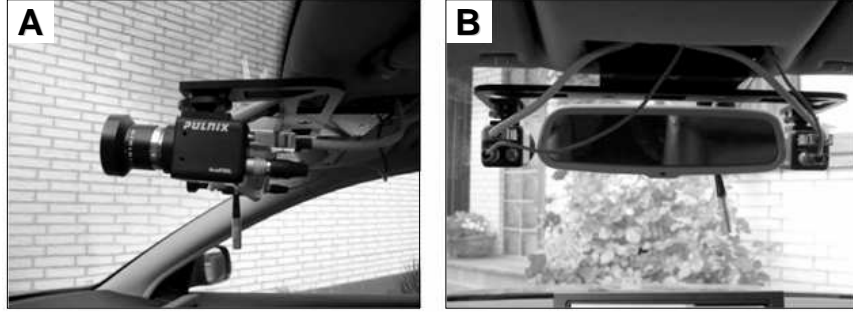
**Table 2.** Video sensor specifications

Sensor parameter	Value
Manufacturer	JAI PULNiX Inc.
Model	TMC-1402CI
Field of View	$53^\circ \times 42.4^\circ$ (horizontal $\times$ vertical)
Used resolution	$1280 \times 1024$
Used frequency	25 fps
Color	RGB Bayer pattern
Interocular distance	330 mm
Focal length	12.5 mm
Optics	Pentax TV lenses

## 2.2 Independent motion stream

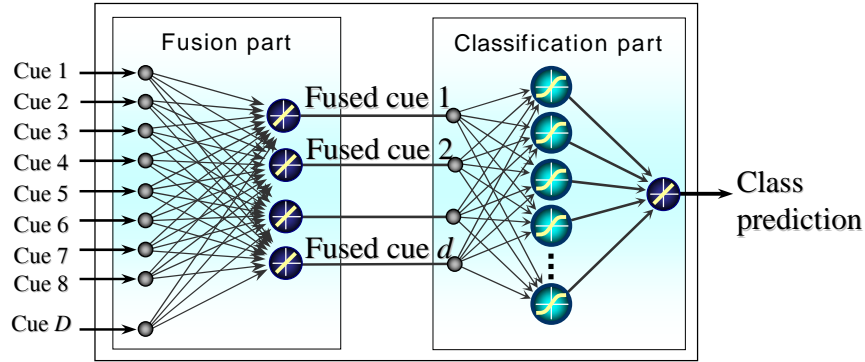
The problem of *independent motion* detection can be defined as the problem of locating objects that move independently from the observer in his field of view. In our case, we build so-called *independent motion maps* where each pixel encodes the likelihood of belonging to an IMO. For each frame we build an independent motion map in two steps: early vision cues extraction and classification.





**Fig. 2.** Setup of the cameras in the car.

As vision cues we consider: *stereo disparity* (three components – for current, previous and next frame), *optical flow* (two components) and *normalized coordinates*<sup>1</sup> (two components). The optic flow and stereo disparity are computed using multiscale phase-based optic flow and stereo disparity algorithms [25, 27]. Unfortunately, there are no possibilities to estimate reliably all these cues for every pixel in the entire frame. This means that the motion stream contains incomplete information, but this gap will be bridged after fusion with the recognition stream.

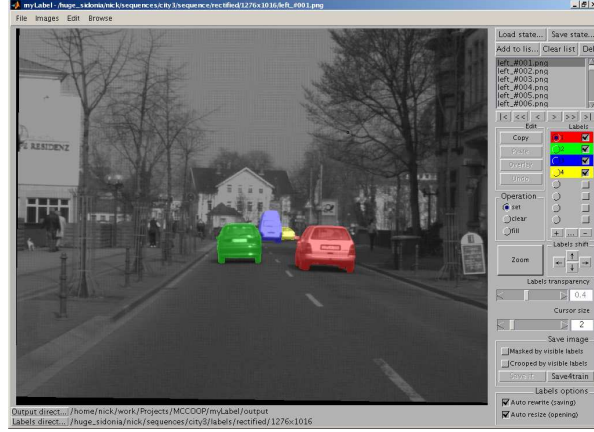


**Fig. 3.** MLP used as classifier in independent motion stream.

We consider each pixel as a multidimensional vector with visual cues as components. We classify all the pixels (which have every component properly defined) in two classes: IMO or background. We have tried a number of setups for classification, but the optimal performance was obtained with a

<sup>1</sup> By a normalized coordinate system on a frame we mean the rectangular coordinate system with origin in the center of the frame, where the upper-left corner is  $(-1, -1)$  and the lower-right corner is  $(1, 1)$ .

multilayered perceptron (MLP) with three layers: a linear (4–8 neurons), a nonlinear layer (8–16 neurons), and one linear neuron as output. For training purposes, we labeled the pixels in every frame of a number of movies into background and different IMOs, using a propriety computer-assisted labeling tool (see Fig. 4).



**Fig. 4.** myLabel – a tool for manual labelling video sequences.

After training, the MLP can be used for building an IMO likelihood map  $I$  for the entire frame:

$$I(x, y) = p(IMO|(x, y)), \quad (1)$$

where  $x, y$  are pixel coordinates. Fig. 5 shows an example of a IMO likelihood map obtained using the proposed approach.

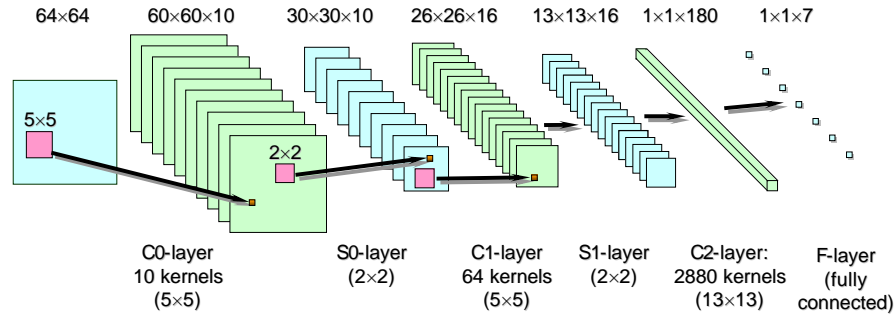
### 2.3 Recognition stream

For the recognition of vehicles and other potentially dangerous objects (such as bicycles and motorcycles, but also pedestrians), we have used a state of the art recognition paradigm – the convolutional network LeNet, proposed by LeCun and colleagues [20]. Modifications of LeNet were successfully applied to generic object recognition [21] and even to obstacle avoidance in an autonomous robot [22]. We have used the CSCSCF configuration of LeNet (see Fig. 6) comprising six layers: three convolutional layers (C0, C1, C2), two subsampling layers (S0, S1) and one fully connected layer (F). As an input, LeNet receives a  $64 \times 64$  gray-scale image. Layer C0 convolves the input with ten  $5 \times 5$  kernels, adds (ten) corresponding biases, and passes the result to a squashing function<sup>2</sup> to obtain ten  $60 \times 60$  feature maps.

<sup>2</sup>  $f(x) = A \tanh(Sx)$ ,  $A = 1.7159$  and  $S = 2/3$  according to [20].



**Fig. 5.** (Left) Frame number 342 of motorway3 sequence. (Right) Matrix  $I$ , output of the motion stream for the same frame. Value  $I(x, y)$  is defined as probability of pixel  $(x, y)$  being part of an IMO.



**Fig. 6.** LeNet – a feed-forward convolutional neural network, used in the recognition stream.

In layer S0, each  $60 \times 60$  map is subsampled to a  $30 \times 30$  map, in such a way that each element of S0 is obtained from a  $2 \times 2$  region of C1 by summing these four elements, by multiplying with a coefficient, adding a bias, and by squashing the end-result. For different S0 elements, the corresponding C1's  $2 \times 2$  regions do not overlap. The S0 layer has ten coefficient-bias couples (one couple for each feature map). Computations in C1 are the same as in C0 with the only difference in the connectivity: each C1 feature map is not obtained by a single convolution, but as a sum of convolutions with a set of previous (S0) maps (see Table 3). Layer S1 subsamples the feature maps of C1 in the same manner as S0 subsamples the feature maps of C0. The final convolutional layer C2 has kernels sized  $13 \times 13$  and 180 feature maps which are fully connected to all 16 S1 feature maps. It means that the number of C2 kernels is  $16 \times 180 = 2880$ , and the corresponding connectivity matrix should have all cells shaded. The output layer consists of seven neurons, which are fully connected to C2's outputs. It means that each neuron in F (corresponding

to a particular class *background*, *cars*, *motorbikes*, *trucks*, *buses*, *bicycles* and *pedestrians*) just squashes the biased weighted sum of all C2's outputs.

		C1 feature maps															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S0 feature maps	0																
	1																
	2																
	3																
	4																
	5																
	6																
	7																
	8																
	9																

**Table 3.** S0-C1 connectivity matrix. A shaded cell which belongs to the  $i$ -th column and  $j$ -th row indicates that the  $j$ -th feature map of S0 participates in the computation of the  $i$ -th feature map of C1. For example, to compute the fourth feature map of C1, one has to find a sum of convolutions of S0 feature maps 0, 8 and 9 with corresponding kernels. The number of kernels in C1 (the number of shaded cells in the table) is 64.

LeNet scans the input image (left frame) in two scales,  $320 \times 256$  and  $640 \times 512$ , with a  $64 \times 64$  sliding window and in 8 and 16 steps, respectively. For each position of the sliding window, we add the output of the class to the corresponding (window) range in a  $320 \times 256$  matrix. In such a way, we obtain seven matrices  $R_0, \dots, R_6$  which, after normalization, are regarded as likelihood maps for the considered classes (see Fig. 7).

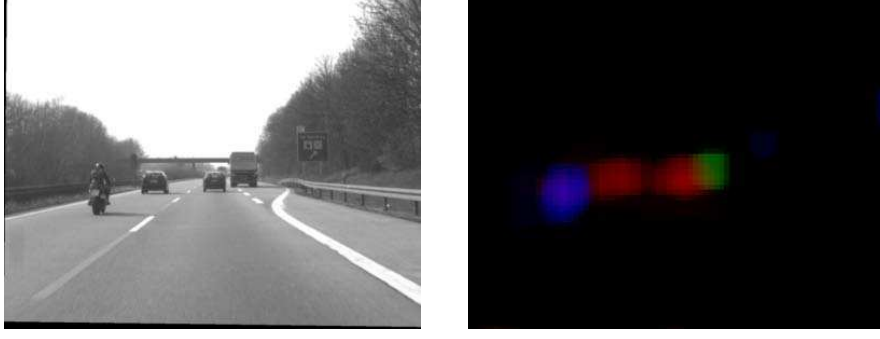
Note that, for further processing, the most important map is  $R_0$ , which corresponds to the background class and the so-called *non-background* map is obtained as  $(1 - R_0)$ . The rest of the maps  $R_1, \dots, R_6$  are responsible only for IMO classification.

## 2.4 Training

For training both streams, we used two rectified stereo video sequences, each consisting of 450 frames. We have labeled IMOs in all left frames of the sequences. These labels were used for training the motion stream classifier.

We have used small batches with the increasing size version of the BFGS Quasi-Newton algorithm for the independent motion classifier training. Samples for each batch were randomly taken from all the frames of all the scenes. Training was stopped after reaching 0.04 (MSE) performance.

To train LeNet, we have prepared a dataset of  $64 \times 64$  grayscale images (approximately 67500 backgrounds, 24500 cars, 2500 motorbikes, 6200 trucks, 1900 bicycles, 78 buses, and 3500 pedestrians). We have doubled the dataset by including horizontally flipped versions of all the samples. Images were



**Fig. 7.** (Left) Frame number 342 of motorway3 sequence. (Right) Output of the recognition stream for the same frame. Here, we used different colors to present different classes: black for background, red for cars, blue for motorcycles and green for trucks.

taken mainly from publicly available object recognition databases (LabelMe<sup>3</sup>, VOC<sup>4</sup>). A stochastic version of the Levenberg-Marquardt algorithm with diagonal approximation of the Hessian [20] was used for LeNet training. Training was stopped after reaching a misclassification rate less than 1.5%. To increase the robustness of the classification, we have run the training procedure several times, every time by adding a small (2%) amount of uniform noise and by randomly changing the intensity (97–103%) of each training sample.

## 2.5 Visual streams fusion

Fusion of the visual streams for a particular frame is achieved in three steps.

1. Intersection of the independent motion map  $I$  with the mask  $M$  of the most probable locations of the IMOs in the frame (see Fig. 8):

$$F_1(x, y) = I(x, y)M(x, y). \quad (2)$$

2. Intersection of the previous result  $F_1$  with the non-background map  $(1 - R_0)$ :

$$F_2(x, y) = F_1(x, y)(1 - R_0(x, y)). \quad (3)$$

3. Intersection of the previous result  $F_2$  with the likelihood maps  $R_1, \dots, R_6$  of each class, which results in six maps  $L_1, \dots, L_6$  (one for each class, except the background):

$$L_k(x, y) = F_2(x, y)R_k(x, y), \quad k = 1, \dots, 6. \quad (4)$$

<sup>3</sup> <http://labelme.csail.mit.edu/>

<sup>4</sup> <http://www.pascal-network.org/challenges/VOC/>

The first step is necessary for rejecting regions of the frame where the appearance of the IMOs is implausible. After the second step we obtain crucial information about regions which have been labeled as non-backgrounds (vehicles or pedestrians) and which, at the same time, contain independently moving objects. This information is represented as the saliency map  $F_2$ , which we will further use for IMO detection/description and in the tracking procedure. The third step provides us the information needed in the classification stage.



**Fig. 8.** Matrix  $M$ , masking regions of possible IMO appearance in a frame.

### 3 IMO Detection and Tracking

For detecting an IMO, we have used a simple technique based on the detection of the local maximas in the maps defined in (3). We have performed a spatio-temporal filtering (i.e. for  $i$ -th frame we apply smoothing of a three-dimensional array – a concatenation of the  $(i-2)$ -th,  $(i-1)$ -th,  $i$ -th,  $(i+1)$ -th and  $(i+2)$ -th two-dimensional maps along the third time-dimension). Then we search for local maximas in the entire ( $i$ -th) filtered frame and consider them as the IMO centers  $\mathbf{x}_k$  for this frame.

For tracking IMOs, we have introduced a parameter called *tracking score*. For a particular IMO, we increase this parameter when, in the next frame, only in a small neighborhood of the IMO center there is a good candidate for the considered IMO in the next frame, namely the IMO with the same class label, and approximately with the same properties (size, distance and relative speed in depth). Otherwise, the tracking score is decreased. An IMO survives while the tracking score is above a fixed threshold. The tracking score works as a momentum and allows the system to keep tracking an IMO even when there are no sufficient data in the next few frames.

## 4 Classification and description of the IMOs

As soon as we are able to detect IMOs, it becomes possible to classify them and to retrieve their properties (size, absolute speed in depth, relative speed in depth, time to contact, absolute acceleration, *etc.*).

We define the *class*  $c_k$  of the  $k$ -th IMO as:

$$c_k = \arg \max_{1 \leq c \leq 6} \{L_c(\mathbf{x}_k)\}, \quad (5)$$

where  $\mathbf{x}_k = (i_k, j_k)$  is the center of the  $k$ -th IMO (in image domain  $D$ ) and  $L_c$  are the maps, defined in (4).

For the  $k$ -th IMO's *size*,  $\sigma_k$ , estimation, we search for a  $\sigma > 0$ , where the first minimum of the function (6) takes place.

$$\Delta_k(\sigma) = \int_D \left| L_{c_k}(\mathbf{x}_k) e^{-\|\mathbf{x}_k - \mathbf{x}\|^2 / \sigma^2} - L_{c_k}(\mathbf{x}) \right| d\mathbf{x}. \quad (6)$$

The IMO's *distance* estimation is a crucial point in the retrieval process. Using an averaged (in a small neighborhood of the IMO's center) disparity and known calibration parameters of the two cameras, we have computed the distance to the IMO. To compensate for instabilities in the distance estimations, we have used a robust linear regression based on the previous five estimates.

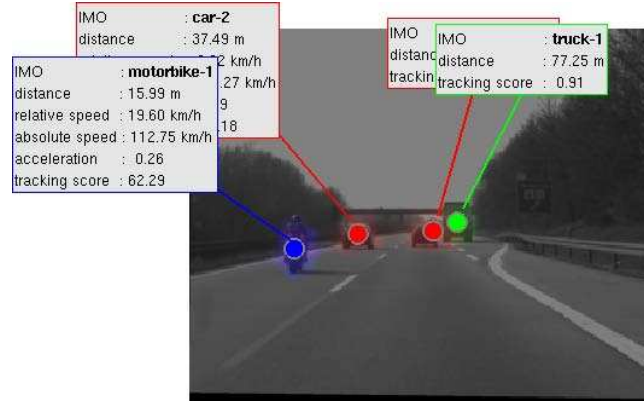
Most of the present-day motor vehicles are being equipped with an increasing number of electronic devices, including control units, sensors, actuators, *etc.* All these devices communicate with each other over a data bus. During recording sessions, we have stored the egospeed provided by test car's speedometer.

The *relative speed in depth*, we estimated as the derivative (with respect to time) of the distance using robust linear regression based on the last five estimations of the distance. To estimate the *time to contact*, we have divided the averaged distance by the averaged relative speed in depth. Using the precise value of the ego-motion speed from the CAN-bus data, and simply by adding it to the relative speed in depth we have also obtained the *absolute speed in depth* of the considered IMO.

The derivative of the absolute speed in depth can be considered as an estimation of the *acceleration* (it is true only in the case when the ego-heading is collinear to the heading of the IMO). An example of IMO tracking and the retrieved properties is shown in Fig. 9.

## 5 LIDAR sensor data processing

The ACC system of the used test car was able to detect and track up to ten obstacles, when in the range of the LIDAR sensor. In addition to position, the ACC can also provide information about relative lateral extent and speed of the tracked obstacle.



**Fig. 9.** Vision-based IMOs detection, classification, description and tracking result.

### 5.1 LIDAR sensor setup

We used data recorded by the test car equipped with the LIDAR sensor manufactured by Hella KGaA Hueck & Co (see Table 4 for specifications). The sensor was mounted in the test car at 30 cm height above ground, with 18 cm from the frontend and 50 cm from the middle of the car to the driver's side (see Fig. 10). The ACC system analyzes raw LIDAR data and tracks up to 10 targets within a distance of up to 150 m. The tracking data are updated and available for recording via the CAN-bus (Flex-ray) every 60 ms. Each tracked target is described by its distance, lateral position (left and right edges), relative velocity and acceleration.

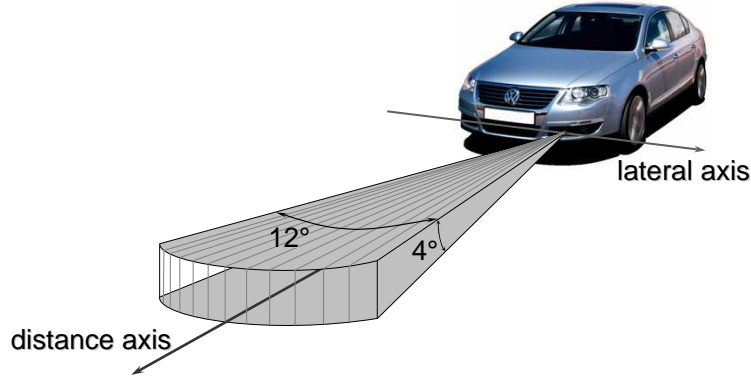
**Table 4.** LIDAR sensor specifications

Sensor parameter	Value
Manufacturer	Hella KGaA Hueck & Co
Model	IDIS 1.0
Field of view	$12^\circ \times 4^\circ$ (horizontal $\times$ vertical)
Range	up to 200 m
Description	12 fixed horizontally distributed beams, each beam observes a $1^\circ \times 4^\circ$ angular cell

### 5.2 Ground plane estimation

The LIDAR provides the depth and lateral position of the detected obstacles. This information is not sufficient for the correct projection of the obstacles onto the video frame. In order to estimate the missing vertical components (in





**Fig. 10.** ACC LIDAR configuration.

the frame domain) of the IMOs we assume that all IMOs are located near the dominant ground plane. Here we use a strong assumption of road planarity, which is not met in all driving scenarios and could introduce bias. However, in our model, the positions of the LIDAR-based obstacles are used only to verify (confirm) vision-based obstacles, so that the bias caused by the non-planarity of the road is to a large extent unimportant.

In order to estimate the ground plane, we estimate the *disparity plane*, then map the set of points from the disparity domain into a 3D world domain, and finally fit a plane through the projected set.

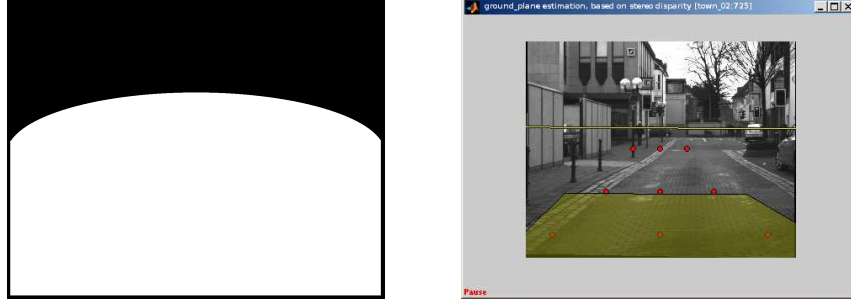
Before the disparity plane estimation, we intersect the disparity map with the predefined road mask (see Fig. 11, left panel). By this step, we filter out the majority of pixels which do not belong to the ground plane and are outliers in the disparity plane linear model:

$$\Delta : D = \alpha x + \beta y + \gamma, \quad (7)$$

where  $(x, y)$  are pixel coordinates and  $D$  is disparity.

The disparity plane parameters  $\alpha, \beta$  and  $\gamma$  are estimated using IRLS (Iteratively Reweighted Least-Squares) with weight function proposed by Beaton and Tukey [1] and tuning parameter  $c = 4.6851$ .

For the ground plane parameters estimation, we choose a set of nine points ( $3 \times 3$  lattice) in the lower half of the frame (see Fig. 11, right panel). Disparities for these points are determined using the estimated disparity plane (7). Given the disparities and camera calibration data, we project the selected points into a 3D world coordinate system. In addition, we add two so-called *stabilization points* which correspond to the points where the front wheels of the test car are supposed to touch the road surface. For the inverse projection of the stabilization points, we use parameters of the *canonic disparity plane*: it is a disparity plane which corresponds to the horizontal ground plane observed by cameras in a quiescent state. The parameters of the canonic disparity plane



**Fig. 11.** (Left) Predefined road mask. (Right) Example of the ground plane estimation. Red points represent points used for ground plane estimation (see text).

and positions of the stabilization points were obtained based on the test car geometry and camera setup position and orientation in the test car. The full set of 11 points is then used for IRLS fitting of the ground plane in a world coordinate system:

$$\pi : aX + bY + cZ + d = 0, \quad (8)$$

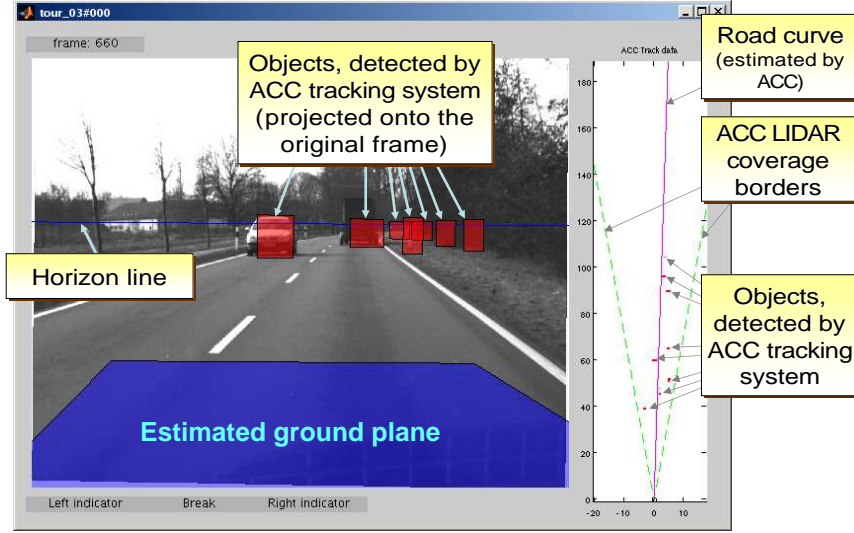
where  $(X, Y, Z)$  are pixel coordinates in the 3D world coordinate system connected to the left camera. Here we assume that  $a^2 + b^2 + c^2 = 1$  (otherwise one can divide all coefficients by  $\sqrt{a^2 + b^2 + c^2}$ ) and  $b > 0$ . In this case vector  $\mathbf{n} = (a, b, c)^T$  represents the normal unity vector of the ground plane and coefficient  $d$  represents the distance from the camera to the ground plane. During the disparity plane estimation, we use the estimation from the previous frame for weight initialization in IRLS; for the first frame, for the same purpose, we use the parameters of the canonic disparity plane. We assume that the ground plane is estimated correctly if the following conditions are met:

$$\|\mathbf{n}_t - \mathbf{n}_0\| < \theta_0 \text{ and } \|\mathbf{n}_t - \mathbf{n}_{t-1}\| < \theta_1, \quad (9)$$

where  $\mathbf{n}_k$  is normal vector for  $k$ -th frame, and  $\mathbf{n}_0$  is canonical normal vector. Thresholds  $\theta_0 = 0.075$  and  $\theta_1 = 0.015$  were chosen empirically. If the estimated ground plane does not satisfy (9), the previous estimation is used.

### 5.3 LIDAR obstacles projection

Projection of the LIDAR-based obstacles into the (left) frame is based on the ground plane position, the obstacle positions, the camera projective matrix (from calibration data) and the position and orientation of the LIDAR sensor with respect to the camera. Only the height of the obstacles is not available. We have set the height of all the obstacles to a fixed value of 1.5 m. The result of the LIDAR obstacles projection is shown in Fig. 12.



**Fig. 12.** ACC obstacles projection. Left part contains the gray-scale version of current frame, overlaid by the horizon line, the ground plane segment and projected ACC (LIDAR) obstacles. Right part represents obstacles 2D range profile, provided by ACC system.

## 6 Vision and LIDAR fusion

The fusion of the vision-based IMOs with LIDAR-based obstacles is based on a simple matching process.

1. For the current IMO  $I_k$ , we look for candidates from the LIDAR obstacles  $O_l$  by means of the high intersection ratio:

$$r_{kl} = \#(I_k \cap O_l) / \#(I_k), \quad (10)$$

where  $\#(\cdot)$  is number of pixels of the set in the brackets. If ratio  $r_{kl} > 0.5$ , then obstacle  $O_l$  is an IMO  $I_k$  candidate and considered for further verification. If all obstacles were rejected, IMO  $I_k$  remains unupdated and process continues from step 4.

2. All the obstacles  $O_{k_m}$  with distances  $d_{k_m}$  satisfying the following condition:

$$\frac{|d_{k_m} - d_k^*|}{d_k^*} > 0.15, \quad (11)$$

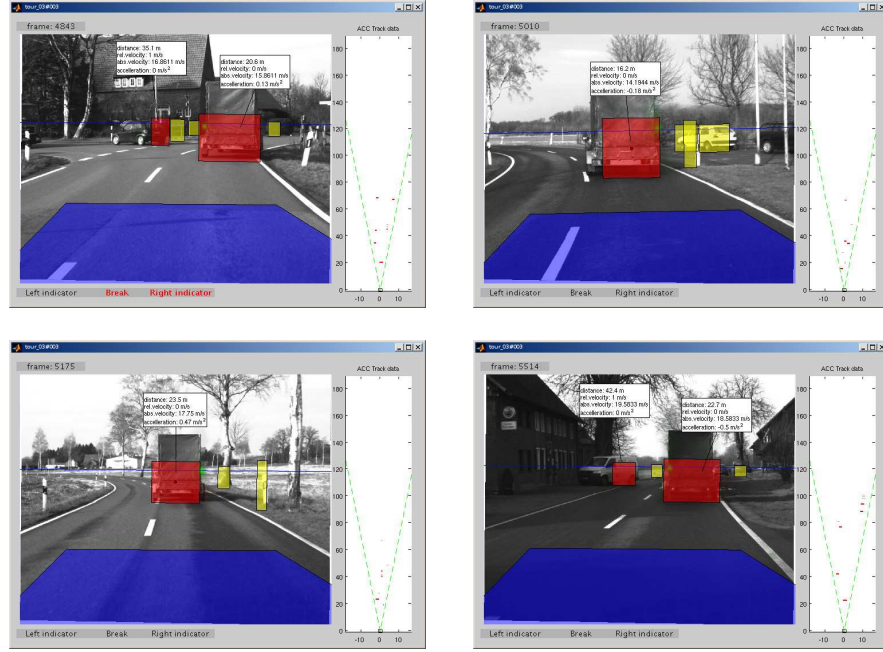
where  $d_k^*$  denotes the distance of the IMO  $I_k$ , are rejected. Like in the previous step, if all obstacles were rejected, IMO  $I_k$  remains unupdated and the process continues from step 4.

3. Among the remaining obstacles, we choose the best matching candidate  $O_{k_i}$  for the IMO  $I_k$  with minimal depth deviation  $|d_{k_i} - d_k^*|$ . Distance, relative velocity and acceleration of IMO  $I_k$  are updated using corresponding

values of the obstacle  $O_{k_i}$ . The absolute velocity of the IMO  $I_k$  is reestimated in accordance with the new value of the relative speed. The obstacle  $O_{k_i}$  is eliminated from the search process. If all the obstacles were rejected, IMO  $I_k$  remains unupdated.

4. The process finishes if all IMOs are checked, otherwise the next IMO is selected for matching and the process continues from step 1.

Some results of the presented fusion are shown on Fig. 13.



**Fig. 13.** Fusion results. Red bars represent detected IMOs, whereas LIDAR obstacles rejected by fusion procedure are shown as yellow bars.

## 7 Conclusions and future steps

A high level sensor fusion model for IMO detection, classification and tracking has been proposed. The model incorporates three independent sensors: vision, LIDAR and speedometer. Vision plays the most important role in the model, whereas LIDAR data are used for confirming the IMO detection and for updating the IMO properties. The speedometer is used only for the IMOs absolute speed in depth estimation.

The existing model is still not a real-time system, but we see a number of ways to increase its speed. Both visual streams of the model have feed-forward architectures, which can be easily implemented in hardware such as Field-Programmable Gate Arrays (FPGAs). Moreover, as far as the streams are independent, they can be implemented as separate FPGAs, working in parallel. In order to speed up the entire model, we propose to switch from LeNet-based object recognition to faster and more task-specific recognition paradigm (e.g. [36] or [23]). Another way to increase the speed of the model could be the transition from an MLP-based fusion of the visual cues to a hard-coded fusion of the visual cues with egomotion (e.g. [24]). As another future step of the model development, we envisage the incorporation of KF-based approaches [14, 35] for IMO tracking.

## 8 Acknowledgements

The work presented in this paper has been supported by the Belgian Fund for Scientific Research – Flanders (G.0248.03), the Excellence Financing program of the K.U.Leuven (EF 2005), the Belgian Fund for Scientific Research – Flanders (G.0248.03, G.0234.04), the Flemish Regional Ministry of Education (Belgium) (GOA 2000/11), the Belgian Science Policy (IUAP P5/04), and the European Commission (NEST-2003-012963, IST-2002-016276, IST-2004-027017)).

## References

- [1] A.E. Beaton and J.W. Tukey. The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. *Technometrics*, 16 (2):147–185, 1974.
- [2] J.C. Becker and A. Simon. Sensor and navigation data fusion for an autonomous vehicle. *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 156–161, 2000.
- [3] M. Bertozzi, A. Broggi, and A. Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 32 (1):1–16, 2000.
- [4] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta. Artificial vision in road vehicles. *Proceedings of the IEEE*, 90(7):1258–1271, 2002.
- [5] C. Blanc, L. Trassoudaine, Y. Le Guilloux, and R. Moreira. Track to track fusion method applied to road obstacle detection. *Proceedings of the Seventh International Conference on Information Fusion*, 2004.
- [6] L. Bombini, P. Cerri, P. Medici, and G. Alessandretti. Radar-vision fusion for vehicle detection.

- [7] N. Chumerin and M.M. Van Hulle. An Approach to On-Road Vehicle Detection, Description and Tracking. *Proceedings of the 2007 17th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, 2007. (in press).
- [8] B.V. Dasarathy. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1):24–38, 1997.
- [9] Y. Fang, I. Masaki, and B. Horn. Depth-based target segmentation for intelligent vehicles: fusion of radar and binocular stereo. *Intelligent Transportation Systems, IEEE Transactions on*, 3(3):196–202, 2002.
- [10] T. Gandhi and M. Trivedi. Vehicle surround capture: Survey of techniques and a novel omni video based approach for dynamic panoramic surround maps. *Intelligent Transportation Systems, IEEE Transactions on*, 7(3):293–308, 2006.
- [11] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.
- [12] U. Handmann, G. Lorenz, T. Schnitger, and W. Seelen. Fusion of different sensors and algorithms for segmentation. *IV'98, IEEE International Conference on Intelligent Vehicles 1998*, pages 499–504, 1998.
- [13] U. Hofmann, A. Rieder, and ED Dickmanns. Radar and vision data fusion for hybrid adaptive cruise control on highways. *Machine Vision and Applications*, 14(1):42–49, 2003.
- [14] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 3, 1997.
- [15] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.
- [16] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21(4):359–381, 2003.
- [17] T. Kato, Y. Ninomiya, and I. Masaki. An obstacle detection method by fusion of radar and motion stereo. *Intelligent Transportation Systems, IEEE Transactions on*, 3(3):182–188, 2002.
- [18] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert. Cooperative Fusion for Multi-Obstacles Detection With Use of Stereovision and Laser Scanner. *Autonomous Robots*, 19(2):117–140, 2005.
- [19] J. Laneurit, C. Blanc, R. Chapuis, and L. Trassoudaine. Multisensorial data fusion for global vehicle and obstacles absolute positioning. *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 138–143, 2003.
- [20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. volume 86, pages 2278–2324, 1998.
- [21] Y. LeCun, F.J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. volume 2, 2004.

- [22] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp. Off-road obstacle avoidance through end-to-end learning. *Advances in neural information processing systems*, 18, 2006.
- [23] B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. *DAGM'04*, pages 145–153, 2004.
- [24] K. Pauwels and M.M. Van Hulle. Segmenting Independently Moving Objects from Egomotion Flow Fields. Isle of Skye, Scotland, 2004.
- [25] K. Pauwels and M.M. Van Hulle. Optic flow from unstable sequences containing unconstrained scenes through local velocity constancy maximization. volume 1, pages 397–406, Edinburgh, 2006.
- [26] C. Pohl. Review article Multisensor image fusion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing*, 19(5):823–854, 1998.
- [27] S.P. Sabatini, G. Gastaldi, F. Solari, J. Diaz, E. Ros, K. Pauwels, M.M. Van Hulle, N. Pugeault, and N. Krueger. Compact and accurate early vision processing in the harmonic space. Barcelona, 2007.
- [28] M.K. Sergi. Bus Rapid Transit Technologies: A Virtual Mirror for Eliminating Vehicle Blind Zones. *University of Minnesota ITS Institute Final Report*, 2003.
- [29] A. Sole, O. Mano, GP Stein, H. Kumon, Y. Tamatsu, A. Shashua, M.E.V.T. Ltd, and I. Jerusalem. Solid or not solid: vision for radar target validation. *Intelligent Vehicles Symposium, 2004 IEEE*, pages 819–824, 2004.
- [30] B. Steux, C. Lurgeau, L. Salesse, and D. Wautier. Fade: a vehicle detection and tracking system featuring monocular color vision and radar data fusion. *Intelligent Vehicle Symposium, 2002. IEEE*, 2, 2002.
- [31] C. Stiller, J. Hipp, C. Rossig, and A. Ewald. Multisensor obstacle detection and tracking. *Image and Vision Computing*, 18(5):389–396, 2000.
- [32] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):694–711, 2006.
- [33] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The Robot that Won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.
- [34] L.G. Ungerleider and T. Pasternak. Ventral and dorsal cortical processing streams. *The Visual Neurosciences*, 1(34):541–562, 2004.
- [35] R. van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, University of Stellenbosch, 2004.
- [36] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. CVPR*, 1:511–518, 2001.
- [37] L. Wald. Some Terms of Reference in Data Fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3), 1999.





---

## Index

ACC, 11  
accelerometer, 1  
Bayer pattern, 4  
canonic disparity plane, 13  
charge-coupled device, CCD, 4  
disparity plane, 13  
extended Kalman filter, EKF, 2  
Field-Programmable Gate Array,  
FPGA, 17  
GPS,DGPS, 1  
ground plane, 13  
gyroscope, 1  
high level fusion, 1  
IMU, 1  
independent motion, 4  
independent motion detection, 4  
independent motion map, 4  
independently moving object, IMO, 1  
intermediate level fusion, 1  
Iteratively Reweighted Least-Squares,  
IRLS, 13  
KF, Kalman filter, 2  
LeNet, 6  
LIDAR, 1, 11  
low level fusion, 1  
multilayered perceptron, MLP, 6  
normalized coordinates, 5  
odometer, 1  
optical flow, 5  
radar, 1  
speedometer, 1  
stereo disparity, 5  
unscented Kalman filter, UKF, 2  
vision cues, 5

## **Appendix D**

### ***First-order and second-order Statistical Analysis of 3D and 2D Structure***

S. Kalkan, F. Wörgötter and N. Krüger

*( Network: Computation in Neural Systems – in press )*

# First-order and Second-order Statistical Analysis of 3D and 2D Image Structure

**S Kalkan<sup>†</sup>, F Wörgötter<sup>†</sup> and N Krüger<sup>‡</sup>**

<sup>†</sup> Bernstein Centre for Computational Neuroscience, University of Göttingen, Germany

<sup>‡</sup> Cognitive Vision Group, University of Southern Denmark, Denmark

E-mail: {sinan,worgott}@chaos.gwdg.de, norbert@mip.sdu.dk

**Abstract.** In the first part of this paper, we analyze the relation between local image structures (i.e., homogeneous, edge-like, corner-like or texture-like structures) and the underlying local 3D structure (represented in terms of continuous surfaces and different kinds of 3D discontinuities) using range data with real-world color images. We find that homogeneous image structures correspond to continuous surfaces, and discontinuities are mainly formed by edge-like or corner-like structures, which we discuss regarding potential computer vision applications and existing assumptions about the 3D world.

In the second part, we utilize the measurements developed in the first part to investigate how the depth at homogeneous image structures is related to the depth of neighbor edges. For this, we first extract the local 3D structure of regularly sampled points, and then, analyze the coplanarity relation between these local 3D structures. We show that the likelihood to find a certain depth at a homogeneous image patch depends on the distance between the image patch and a neighbor edge. We find that this dependence is higher when there is a second neighbor edge which is coplanar with the first neighbor edge. These results allow deriving statistically based prediction models for depth interpolation on homogeneous image structures.

Submitted to: *Network: Comput. Neural Syst.*

## 1. Introduction

Depth estimation relies on the extraction of 3D structure from 2D images which is realized by a set of inverse problems including structure from motion, stereo vision, shape from shading, linear perspective, texture gradients and occlusion [Bruce et al., 2003]. In methods which make use of multiple views (*i.e.*, stereo and structure from motion), correspondences between different 2D views of the scene are required. In contrast, monocular or pictorial cues such as shape from shading, texture gradients or linear perspective use statistical and geometrical relations within one image to make statements about the underlying 3D structure.

Many surfaces have only weak texture or no texture at all, and as a consequence, the correspondence problem is very hard or not at all resolvable for these surfaces. Nevertheless, humans are able to reconstruct the 3D information for these surfaces, too. This gives rise to the assumption that in the human visual system, an interpolation process is realised that, starting with the local analysis of edges, corners and textures, computes depth also in areas where correspondences cannot easily be found.

Processing of depth in the human visual system starts with the processing of local image structures (such as edge-like structures, corner-like structures and textures) in V1 [Hubel and Wiesel, 1969, Gallant et al., 1994, Lee et al., 1998]. These structures (called 2D structures in the rest of the paper) are utilized in stereo vision, depth from motion, depth from texture gradients and other depth cues, which are localized in different parts of the brain, starting from V1 and involving V2, V3, V4 and MT (see, *e.g.*, [Serenio et al., 2002]).

There exists good evidence that depth cues which are not directly based on correspondences evolve rather late in the development of the human visual system. For example, pictorial depth cues are made use of only after approximately 6 months [Kellman and Arterberry, 1998]. This indicates that experience may play an important role in the development of these cues, *i.e.*, that we have to understand depth perception as a statistical learning problem [Knill and Richards, 1996, Rao et al., 2002, Purves and Lotto, 2002]. A step towards such an understanding is the investigation and use of the statistical relations between the local 2D structures and the underlying 3D structure for each of these depth cues [Knill and Richards, 1996, Rao et al., 2002, Purves and Lotto, 2002].

With the notion that the human visual system is adapted to the statistics of the environment [Brunswik and Kamiya, 1953, Knill and Richards, 1996, Krueger, 1998, Olshausen and Field, 1996, Rao et al., 2002, Purves and Lotto, 2002, Simoncelli, 2003] and its successful applications to grouping, object recognition and stereo [Elder and Goldberg, 2002, Elder et al., 2003, Pugeault et al., 2004, Zhu, 1999], the analysis and the usage of natural image statistics have become an important focus of vision research. Moreover, with the advances in technology, it has been also possible to analyze the 3D world using 3D range scanners [Howe and Purves, 2004, Huang et al., 2000, Potetz and Lee, 2003, Yang and Purves, 2003].

In this paper, we analyze first-order and second-order relations<sup>‡</sup> between 2D and 3D

<sup>‡</sup> In this paper, a relation is first-order if it involves two entities and an event between them. Analogously, a relation is second-order if there are three entities and (at least) two events between them.

structures extracted from chromatic 3D range data§. For the first-order analysis, we investigate the relation between local 2D structures (*i.e.*, homogeneous, edge-like, corner-like or texture-like structures) and the underlying local 3D structure. As for the second-order analysis, we investigate the relation between the depth at homogeneous 2D structures and the depth at the bounding edges.

There have been only a few studies that have analyzed the 3D world from range data [Howe and Purves, 2004, Huang et al., 2000, Potetz and Lee, 2003, Yang and Purves, 2003], and these works have only been first-order. In [Yang and Purves, 2003], the distribution of roughness, size, distance, 3D orientation, curvature and independent components of surfaces was analyzed. Their major conclusions were: (1) local 3D patches tend to be saddle-like, and (2) natural scene geometry is quite regular and less complex than luminance images. In [Huang et al., 2000], the distribution of 3D points was analyzed using co-occurrence statistics and 2D and 3D joint distributions of Haar filter reactions. They showed that range images are much simpler to analyze than optical images and that a 3D scene is composed of piecewise smooth regions. In [Potetz and Lee, 2003], the correlation between light intensities of the image data and the corresponding range data as well as surface convexity were investigated. They could justify the event that brighter objects are closer to the viewer, which is used by shape from shading algorithms in estimating depth. In [Howe and Purves, 2002, Howe and Purves, 2004], range image statistics were analyzed for explanation of several visual illusions.

Our first-order analysis differs from these works. For 2D local image patches, existing studies have only considered light intensity. As for 3D local patches, the most complex considered representation has been the curvature of the local 3D patch. In this work, however, we create a higher-order representation of the 2D local image patches and the 3D local patches; we represent 2D local image patches using homogeneous, edge-like, corner-like or texture-like structures, and 3D local patches using continuous surfaces and different kinds of 3D discontinuities. By this, we relate established local 2D structures to their underlying 3D structures.

For the first-order analysis, we compute the conditional likelihood  $P(\text{3D Structure} \mid \text{2D Structure})$ , by creating 2D and 3D representations of the local structure. Using this likelihood, we quantify some assumptions made by the studies that reconstruct the 3D world from dense range data. For example, we will show that the depth distribution varies significantly for different visual features, and we will quantify already established inter-dependencies such as 'no news is good news' [Grimson, 1983]. This work also supports the understanding of how intrinsic properties of 2D–3D relations can be used for the reconstruction of depth, for example, by using statistical priors in the formalisation of depth cues.

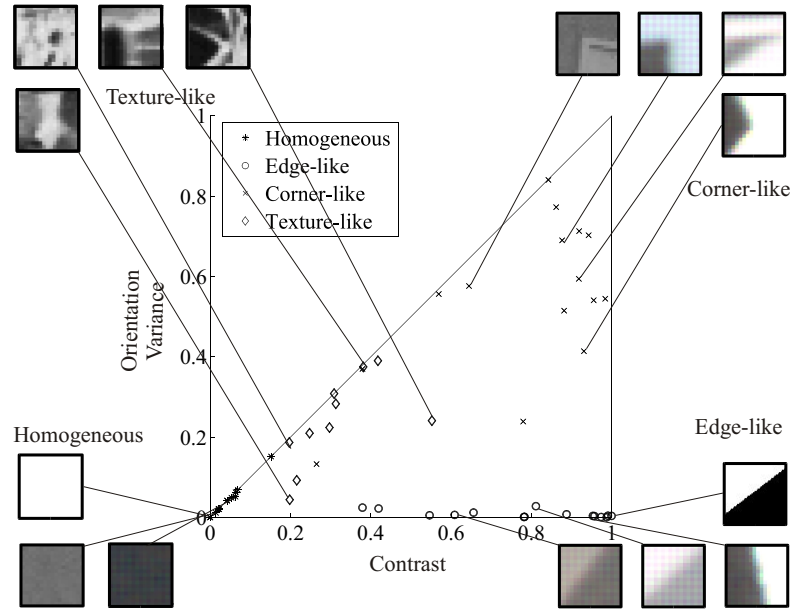
For the second-order analysis, given two proximate co-planar edges, we compute the 'likelihood field' of finding co-planar surface patches which project as homogeneous 2D structures in the 2D image. This likelihood field is similar to the 'association field' [Field et al., 1993] which is a likelihood field also based on natural image statistics.

§ In this paper, chromatic 3D range data means range data which has associated real-world color information. The color information is acquired using a digital camera which is calibrated with the range scanner.

The 'likelihood field' which we compute provides important information about (1) the predictability of depth at homogeneous 2D structures using the depth available at the bounding edges and (2) the relative complexity of 3D geometric structure compared to the complexity of local 2D structures.

The paper is organized as follows: In sections 2 and 3, we define the types of local 2D structures and local 3D structures and how we extract them for our analysis. In section 4, we analyze the relation between the local 2D and 3D structures, and discuss the results. In section 5, we present our methods for analyzing the second-order relation between the homogeneous 2D structures and bounding edge structures, and discuss the results. Finally, we conclude the paper in section 6 with a discussion.

## 2. Local 2D Structures



**Figure 1.** How a set of 54 patches map to the different areas of the intrinsic dimensionality triangle. Some examples from these patches are also shown. The horizontal and vertical axes of the triangle denote the contrast and the orientation variances of the image patches, respectively.

We distinguish between the following local 2D structures (examples of each structure is given in figure 1):

- **Homogeneous 2D structures:** Homogeneous 2D structures are signals of uniform intensities, and they are not much made use of in the human visual system because retinal ganglion cells give only weak sustained responses and adapt quickly at homogeneous intensities [Bruce et al., 2003].
- **Edge-like 2D structures:** Edges are low-level structures which constitute the boundaries between homogeneous or texture-like signals. Detection of edge-like

structures in the human visual system starts with orientation sensitive cells in V1 [Hubel and Wiesel, 1969], and biological and machine vision systems depend on their reliable extraction and utilization [Marr, 1982, Koenderink and Dorn, 1982].

- Corner-like 2D structures: Corners $\parallel$  are image patches where two or more edge-like structures with significantly different orientations intersect (see, *e.g.*, [Guzman, 1968, Rubin, 2001] for their importance in vision). It has been suggested that the human visual system makes use of them for different tasks like recovery of surface occlusion [Guzman, 1968, Rubin, 2001] and shape interpretation [Malik, 1987].
- Texture-like 2D structures: Although there is not a widely-agreed definition, textures are often defined as signals which consist of repetitive, random or directional structures (for their analysis, extraction and importance in vision, see *e.g.*, [Tuceryan and Jain, 1998]). Our world consists of textures on many surfaces, and the fact that we can reliably reconstruct the 3D structure from any textured environment indicates that human visual system makes use of and is very good at the analysis and the utilization of textures. In this paper, we define texture as 2D structures which have low spectral energy and a lot of orientation variance (see figure 1 and section 2.1).

It is locally hard to distinguish between these 'ideal' cases, and there are 2D structures that carry mixed properties of these 'ideal' cases. The classification of the features outlined above is a discrete one. However, a discrete classification may cause problems as the inherent properties of the "mixed" structures are lost in the discretization process. Instead, in this paper, we make use of a continuous scheme which is based on the concept of intrinsic dimensionality (see section 2.1 for more details).

### 2.1. Detection of Local 2D Structures

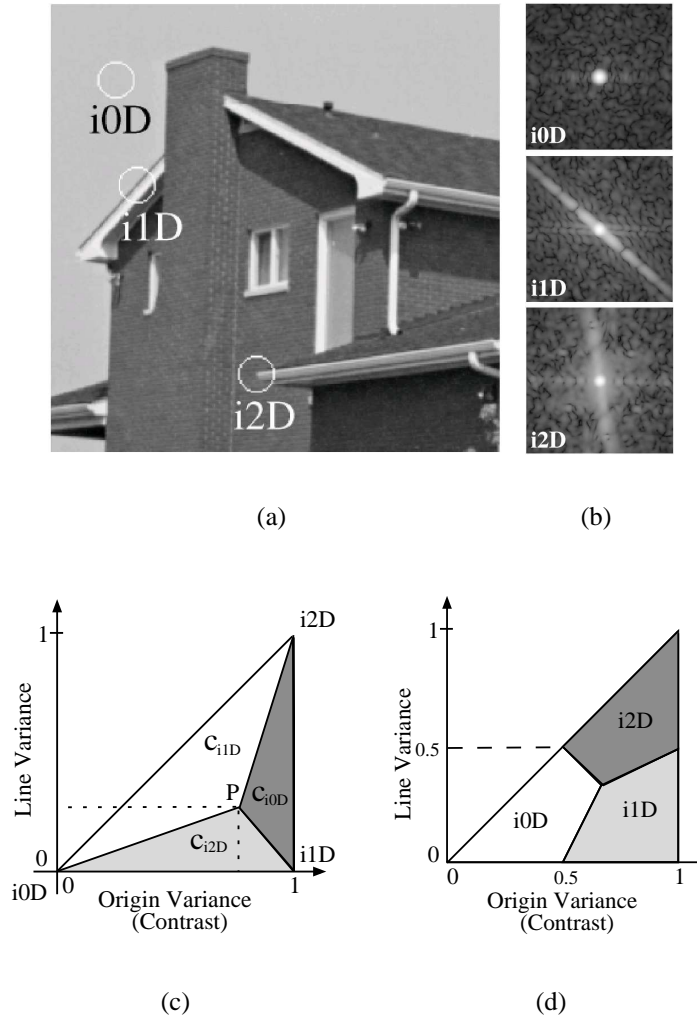
In image processing, intrinsic dimensionality (iD) was introduced by [Zetsche and Barth, 1990] and was used to formalize a *discrete distinction* between edge-like and junction-like structures. This corresponds to a classical interpretation of local 2D structures in computer vision.

Homogeneous, edge-like and junction-like structures are respectively classified by iD as *intrinsically zero dimensional (i0D)*, *intrinsically one dimensional (i1D)* and *intrinsically two dimensional (i2D)*.

When looking at the spectral representation of a local image patch (see figure 2(a,b)), we see that the energy of an i0D signal is concentrated in the origin (figure 2(b)-top), the energy of an i1D signal is concentrated along a line (figure 2(b)-middle) while the energy of an i2D signal varies in more than one dimension (figure 2(b)-bottom).

It has been shown in [Felsberg and Krüger, 2003, Krüger and Felsberg, 2003] that the structure of the iD can be understood as a triangle that is spanned by two measures: origin variance (*i.e.*, contrast) and line variance. Origin variance describes the deviation of the energy from a concentration at the origin while line variance describes the deviation from a line structure (see figure 2(b) and 2(c)); in other words, origin variance measures non-homogeneity

$\parallel$  In this paper, for the sake of simplicity, junctions are called corners, too.



**Figure 2.** Illustration of iD (Sub-figures (a,b) taken from [Felsberg and Krüger, 2003]). **(a)** Three image patches for three different intrinsic dimensions. **(b)** The 2D spatial frequency spectra of the local patches in (a), from top to bottom:  $i0D$ ,  $i1D$ ,  $i2D$ . **(c)** The topology of iD. Origin variance is variance from a point, i.e., the origin. Line variance is variance from a line, measuring the junction-ness of the signal.  $c_{iND}$  for  $N = 0, 1, 2$  stands for confidence for being  $i0D$ ,  $i1D$  and  $i2D$ , respectively. Confidences for an arbitrary point  $P$  is shown in the figure which reflect the areas of the sub-triangles defined by  $P$  and the corners of the triangle. **(d)** The decision areas for local 2D structures.

of the signal whereas the line variance measures the junctionness. The corners of the triangle then correspond to the 'ideal' cases of iD. The surface of the triangle corresponds to signals that carry aspects of the three 'ideal' cases, and the distance from the corners of the triangle indicates the similarity (or dissimilarity) to *ideal*  $i0D$ ,  $i1D$  and  $i2D$  signals.

The triangular structure of the intrinsic dimension is counter-intuitive, in the first place, since it realizes a two-dimensional topology in contrast to a linear one-dimensional structure that is expressed in the discrete counting 0, 1 and 2. As shown in [Krüger and Felsberg, 2003, Felsberg and Krüger, 2003], this triangular interpretation allows for a *continuous formulation*





**Figure 3.** Computed iD for the image in figure 2, black means zero and white means one. From left to right:  $c_{i0D}$ ,  $c_{i1D}$ ,  $c_{i2D}$  and highest confidence marked in gray, white and black for i0D, i1D and i2D, respectively.

of iD in terms of 3 confidences assigned to each discrete case. This is achieved by first computing two measurements of origin and line variance which define a point in the triangle (see figure 2(c)). The bary-centric coordinates (see, e.g., [Coxeter, 1969]) of this point in the triangle directly lead to a definition of three confidences that add up to one:

$$c_{i0D} = 1 - x, c_{i1D} = x - y, c_{i2D} = y. \quad (1)$$

These three confidences reflect the volume of the areas of the three sub-triangles which are defined by the point in the triangle and the corners of the triangle (see figure 2(c)). For example, for an arbitrary point  $P$  in the triangle, the area of the sub-triangle  $i0D-P-i1D$  denotes the confidence for i2D as shown in figure 2(c). That leads to the decision areas for i0D, i1D and i2D as seen in figure 2(d). See appendix [Felsberg and Krüger, 2003, Krüger and Felsberg, 2003] for more details.

For the example image in figure 2, computed iD is given in figure 3.

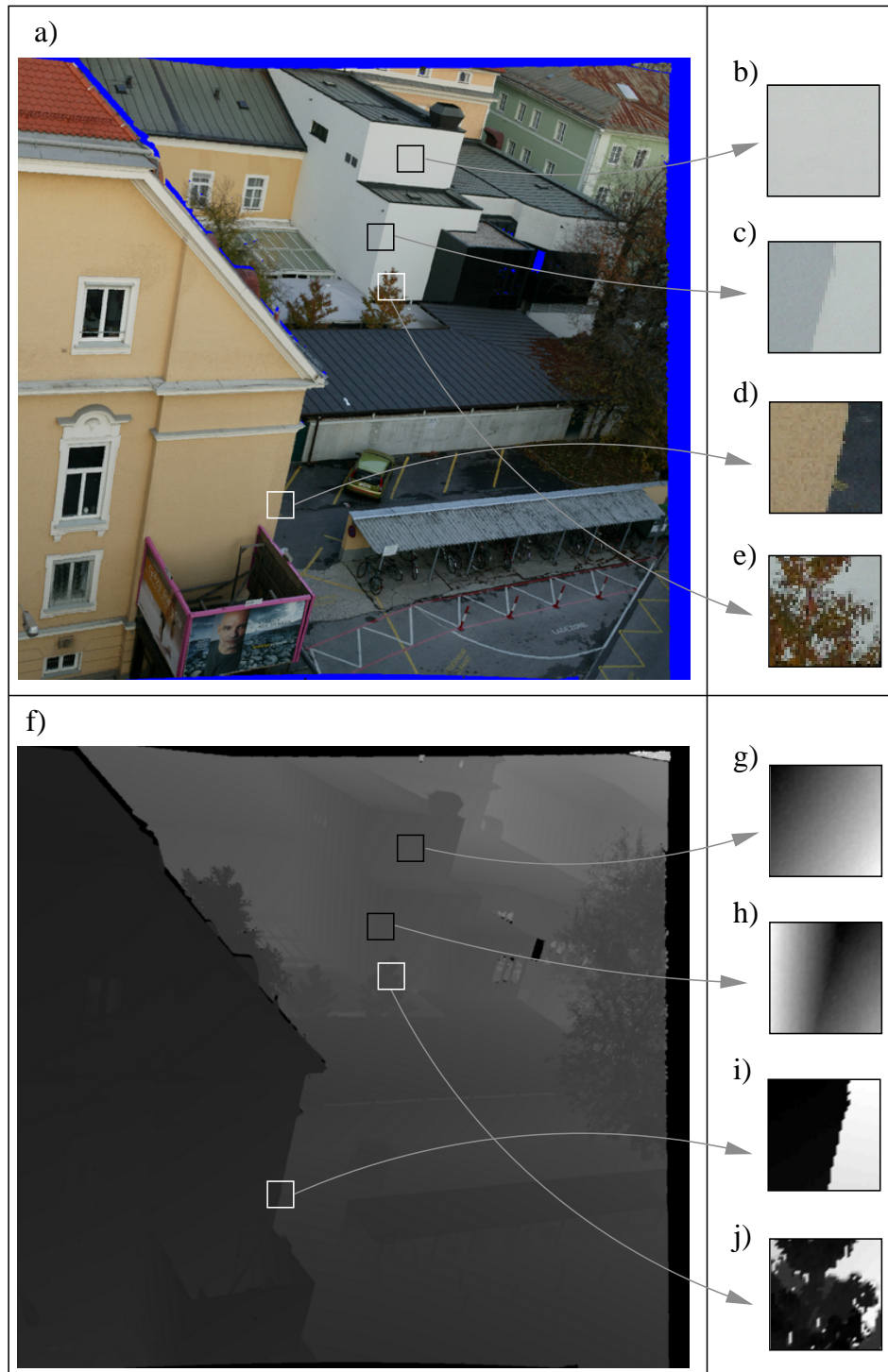
Figure 1 shows how a set of example local 2D structures map on to it. In figure 1, we see that different visual structures map to different areas in the triangle. A detailed analysis of how 2D structures are distributed over the intrinsic dimensionality triangle and how some visual information depends on this distribution can be found in [Kalkan et al., 2005].

### 3. Local 3D Structures

To our knowledge, there does not exist a systematic and agreed classification of local 3D structures like there is for 2D local structures (*i.e.*, homogeneous structures, edges, corners and textures). Intuitively, the 3D world consists of continuous surface patches and different kinds of 3D discontinuities. During the imaging process (through the lenses of the camera or the eye), 2D local structures are generated by these 3D structures together with the illumination and the reflectivity of the environment.

With this intuition, any 3D scene can be decomposed geometrically into surfaces and 3D discontinuities. In this context, the local 3D structure of a point can be a:

- **Surface Continuity:** The underlying 3D structure can be described by one surface whose normal does not change or changes smoothly (see figure 4(a)).



**Figure 4.** Illustration of the types of 3D discontinuities. (a) 2D image. (b) Continuity. (c) Orientation discontinuity. (d) Gap discontinuity. (e) Irregular gap discontinuity. (f)-(j) The range images corresponding to (a)-(e). Note that the range images are scaled independently for better visibility.



**Figure 5.** 10 of the 20 3D data sets used in the analysis. The points without range information are marked in blue. The gray image shows the range data of the top-left scene. The horizontal and the vertical resolutions of the scenes respectively have the following ranges: [512-2048] and [390-2290]. The average resolution of the scenes is 1140x1001.

- **Regular Gap discontinuity:** Regular gap discontinuities are occlusion boundaries, whose underlying 3D structure can be described by a small set of surfaces with a significant depth difference. The 2D and 3D views of an example gap discontinuity are shown in figure 4(d).
- **Irregular Gap discontinuity:** The underlying 3D structure shows high depth-variation that can not be described by two or three surfaces. An example of an irregular gap discontinuity is shown in figure 4(e).
- **Orientation Discontinuity:** The underlying 3D structure can be described by two surfaces with significantly different 3D orientations that meet at the center of the patch. This type of discontinuity is produced by a change in 3D orientation rather than a gap between surfaces. An example for this type of discontinuity is shown in figure 4(c).

One interesting example is 3D corners of, for example, a cube. 3D corners would be classified as regular gap discontinuities or orientation discontinuities, depending on the view. If the image patch includes parts of the background objects, then there is a gap discontinuity, and the 3D corner would be classified as a gap discontinuity. If, however, the camera centers the corner so that all the adjacent edges of the cube are visible and no parts of other objects are visible, then the 3D corner would be an orientation discontinuity.

### 3.1. Detection of Local 3D Structures

In this subsection, we define our measures for the three kinds of discontinuities that we described above; namely, gap discontinuity, irregular gap discontinuity and orientation discontinuity. The measures for gap discontinuity, irregular gap discontinuity and orientation

discontinuity of a patch  $P$  will be denoted by  $\mu_{GD}(P)$ ,  $\mu_{IGD}(P)$  and  $\mu_{OD}(P)$ , respectively. The reader who is not interested in the technical details can jump directly to section 4.

3D discontinuities are detected in studies which involve range data processing, using different methods and under different names like two-dimensional discontinuous edge, jump edge or depth discontinuity for gap discontinuity; and, two-dimensional corner edge, crease edge or surface discontinuity for orientation discontinuity [Bolle and Vemuri, 1991, Hoover et al., 1996, Shirai, 1987].

In our analysis, we used chromatic range data of outdoor scenes which were obtained from Riegl UK Ltd. (<http://www.riegl.co.uk/>). There were 20 scenes in total, 10 of which are shown in figure 5. The range of an object which does not reflect the laser beam back to the scanner or is out of the range of the scanner cannot be measured. These points are marked with blue in figure 5 and are not processed in our analysis. The horizontal and the vertical resolutions of the scenes respectively have the following ranges: [512-2048] and [390-2290]. The average resolution of the scenes is 1140x1001.

### 3.1.1. Measure for Gap Discontinuity: $\mu_{GD}$

Gap discontinuities can be measured or detected in a similar way than edges in 2D images; edge detection processes RGB-coded 2D images while for a gap discontinuity, one needs to process XYZ-coded 2D images ¶. In other words, gap discontinuities can be measured or detected by taking the second order derivative of XYZ values [Shirai, 1987].

Measurement of a gap discontinuity is expected to operate on both the horizontal and the vertical axes of the 2D image; that is, it should be a two dimensional function. The alternative is to discard the topology and do an 'edge-detection' in sorted XYZ values, *i.e.*, to operate as a one-dimensional function. Although we are not aware of a systematic comparison of the alternatives, for our analysis and for our data, the topology-discarding gap discontinuity measurement captured the underlying 3D structure better (of course, qualitatively, *i.e.*, by visual inspection). Therefore, we have adopted the topology-discarding gap discontinuity measurement in the rest of the paper.

For an image patch  $P$  of size  $N \times N$ , let,

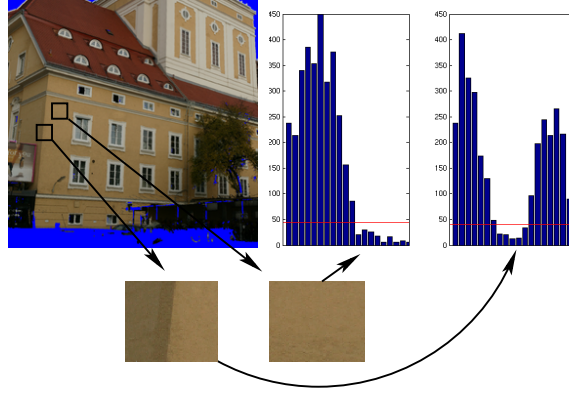
$$\begin{aligned}\mathcal{X} &= \text{ascending\_sort}(\{X_i \mid i \in P\}), \\ \mathcal{Y} &= \text{ascending\_sort}(\{Y_i \mid i \in P\}), \\ \mathcal{Z} &= \text{ascending\_sort}(\{Z_i \mid i \in P\}),\end{aligned}\tag{2}$$

and also, for  $i = 1, \dots, (N \times N - 2)$ ,

$$\begin{aligned}\mathcal{X}^\Delta &= \{ | (\mathcal{X}_{i+2} - \mathcal{X}_{i+1}) - (\mathcal{X}_{i+1} - \mathcal{X}_i) | \}, \\ \mathcal{Y}^\Delta &= \{ | (\mathcal{Y}_{i+2} - \mathcal{Y}_{i+1}) - (\mathcal{Y}_{i+1} - \mathcal{Y}_i) | \}, \\ \mathcal{Z}^\Delta &= \{ | (\mathcal{Z}_{i+2} - \mathcal{Z}_{i+1}) - (\mathcal{Z}_{i+1} - \mathcal{Z}_i) | \},\end{aligned}\tag{3}$$

where  $\mathcal{X}_i, \mathcal{Y}_i, \mathcal{Z}_i$  represents 3D coordinates of pixel  $i$ . Equation 3 takes the absolute value of the  $[+1, -2, +1]$  operator.

¶ Note that XYZ and RGB coordinate systems are not the same. However, detection of gap discontinuity in XYZ coordinates can be assumed to be a special case of edge detection in RGB coordinates.



**Figure 6.** Example histograms and the number of clusters that the function  $\psi(S)$  computes.  $\psi(S)$  finds one cluster in the left histogram and two clusters in the right histogram. Red line marks the threshold value of the function. X axis denotes the values for 3D orientation differences.

The sets  $\mathcal{X}^\Delta, \mathcal{Y}^\Delta$  and  $\mathcal{Z}^\Delta$  are the measurements of the jumps (*i.e.*, second order differentials) in the sets  $\mathcal{X}, \mathcal{Y}$  and  $\mathcal{Z}$ , respectively. A gap discontinuity can be defined simply as a measure of these jumps in these sets. In other words:

$$\mu_{GD}(P) = \frac{h(\mathcal{X}^\Delta) + h(\mathcal{Y}^\Delta) + h(\mathcal{Z}^\Delta)}{3}, \quad (4)$$

where the function  $h : \mathcal{S} \rightarrow [0, 1]$  over the set  $\mathcal{S}$  measures the homogeneity of its argument set (in terms of its 'peakiness') and is defined as follows:

$$h(\mathcal{S}) = \frac{1}{\#(\mathcal{S})} \times \sum_{i \in \mathcal{S}} \frac{s_i}{\max(\mathcal{S})}, \quad (5)$$

where  $\#(\mathcal{S})$  is the number of the elements of  $\mathcal{S}$ , and  $s_i$  is the  $i^{th}$  element of the set  $\mathcal{S}$ . Note that as a homogeneous set (*i.e.*, a non-gap discontinuity)  $\mathcal{S}$  produces a high  $h(\mathcal{S})$  value, a gap discontinuity causes a low  $\mu_{GD}$  value. Figure 8(c) shows the performance of  $\mu_{GD}$  on one of our scenes shown in figure 5.

It is known that derivatives like in equations 2 and 3 are sensitive to noise. Gaussian-based functions could be employed instead. In this paper, we chose simple derivatives for their faster computation times, and instead employed a more robust processing stage (*i.e.*, analyzing the uniformity of the distribution of derivatives) to make the measurement more robust to noise. As shown in figure 8(c), this method can capture the underlying 3D structure well.

### 3.1.2. Measure for Orientation Discontinuity: $\mu_{OD}$

The orientation discontinuity of a patch  $P$  can be detected or measured by taking the 3D orientation difference between the surfaces that meet in  $P$ . If the size of the patch  $P$  is small enough, the surfaces can be, in practice, approximated by 2-pixel wide unit planes<sup>+</sup>.

<sup>+</sup> Note that using bigger planes have the disadvantage of losing accuracy in positioning which is very crucial for the current analysis.

The histogram of the 3D orientation differences between every pair of unit planes forms one cluster for continuous surfaces and two clusters for orientation discontinuities.

For an image patch  $P$  of size  $N \times N$  pixels, the orientation discontinuity measure is defined as:

$$\mu_{OD}(P) = \psi(H^n(\{\alpha(i, j) \mid i, j \in \text{planes}(P), i \neq j\})), \quad (6)$$

where  $H^n(S)$  is a function which computes the  $n$ -bin histogram of its argument set  $S$ ;  $\psi(S)$  is a function which finds the number of clusters in  $S$ ;  $\text{planes}(P)$  is a function which fits 2-pixel-wide unit planes to 1-pixel apart points in  $P$  using Singular Value Decomposition\*; and,  $\alpha(i, j)$  is the angle between planes  $i$  and  $j$ .

For a histogram  $H$  of size  $N_H$ , the number of clusters is given by:

$$\psi(S) = \frac{\sum_{i=1}^{N_H+1} \text{neq}([H_i > \max(H)/10], [H_{i-1} > \max(H)/10])}{2}, \quad (7)$$

where the function  $\text{neq}$  returns 1 if its parameters are not equal and returns 0, otherwise;  $H_i$  represents the  $i^{\text{th}}$  element of the histogram  $H$ ;  $H_0$  and  $H_{N_H+1}$  are defined as zero; and,  $\max(H)/10$  is an empirically set threshold. Figure 6 shows two example clusters for a continuous surface and an orientation discontinuity.

Figure 8(d) shows the performance of  $\mu_{OD}$  on one of our scenes shown in figure 5.

### 3.1.3. Measure for Irregular Gap Discontinuity: $\mu_{IGD}$

Irregular gap discontinuity of a patch  $P$  can be measured using the observation that an irregular-gap discontinuous patch in a real image usually consists of small surface fragments with different 3D orientations. Therefore, the spread of the 3D orientation histogram of a patch  $P$  can measure the irregular gap discontinuity of  $P$ .

Similar to the measure for orientation discontinuity defined in sections 3.1.1 and 3.1.2, the histogram of the differences between the 3D orientations of the unit planes (which are of 2 pixels wide) is analyzed. For an image patch  $P$  of size  $N \times N$  pixels, the irregular gap discontinuity measure is defined as:

$$\mu_{IGD}(P) = h(H^n(\{\alpha(i, j) \mid i, j \in \text{planes}(P), i \neq j\})), \quad (8)$$

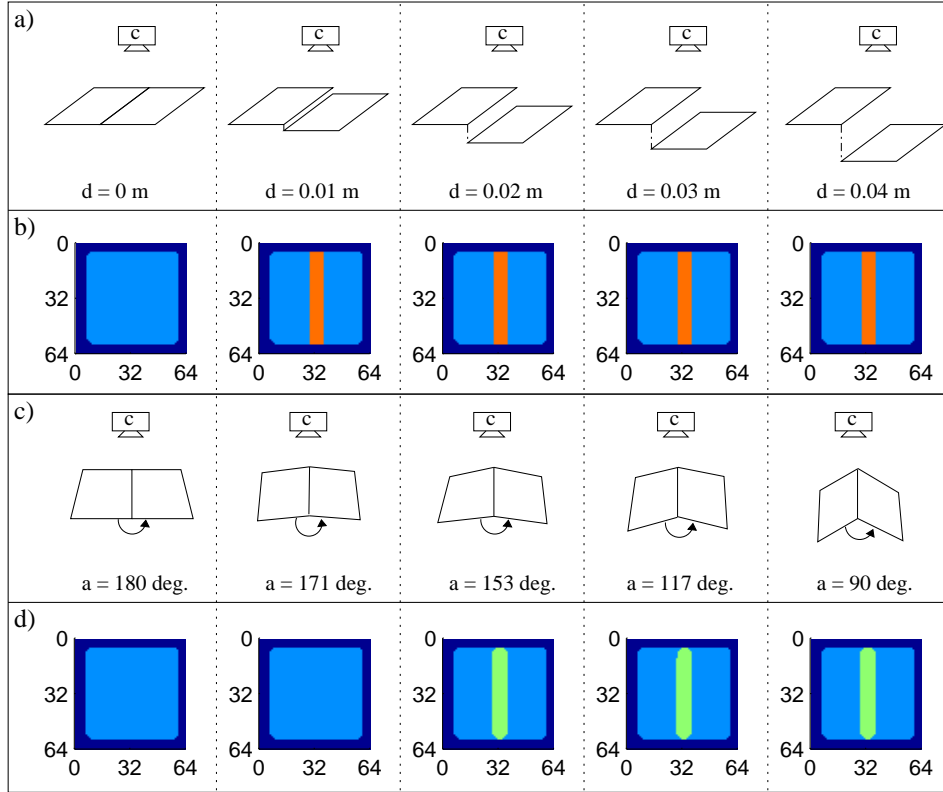
where  $\text{planes}(P)$ ,  $\alpha(i, j)$ ,  $H^n(S)$  and  $h(S)$  are as defined in section 3.1.2. Figure 8(e) shows the performance of  $\mu_{IGD}$  on one of our scenes shown in figure 5.

### 3.1.4. Combining the Measures

The relation between the measurements and the types of the 3D discontinuities are outlined in table 1 which entails that an image patch  $P$  is:

- gap discontinuous if  $\mu_{GD}(P) < T_g$  and  $\mu_{IGD}(P) < T_{ig}$ ,
- irregular-gap discontinuous if  $\mu_{GD}(P) < T_g$  and  $\mu_{IGD}(P) > T_{ig}$ ,
- orientation discontinuous if  $\mu_{GD}(P) \geq T_g$  and  $\mu_{OD} > 1$ ,

\* Singular Value Decomposition is a standard technique for fitting planes to a set of points. It finds the perfectly fitting plane if it exists; otherwise, it returns the least-squares solution.



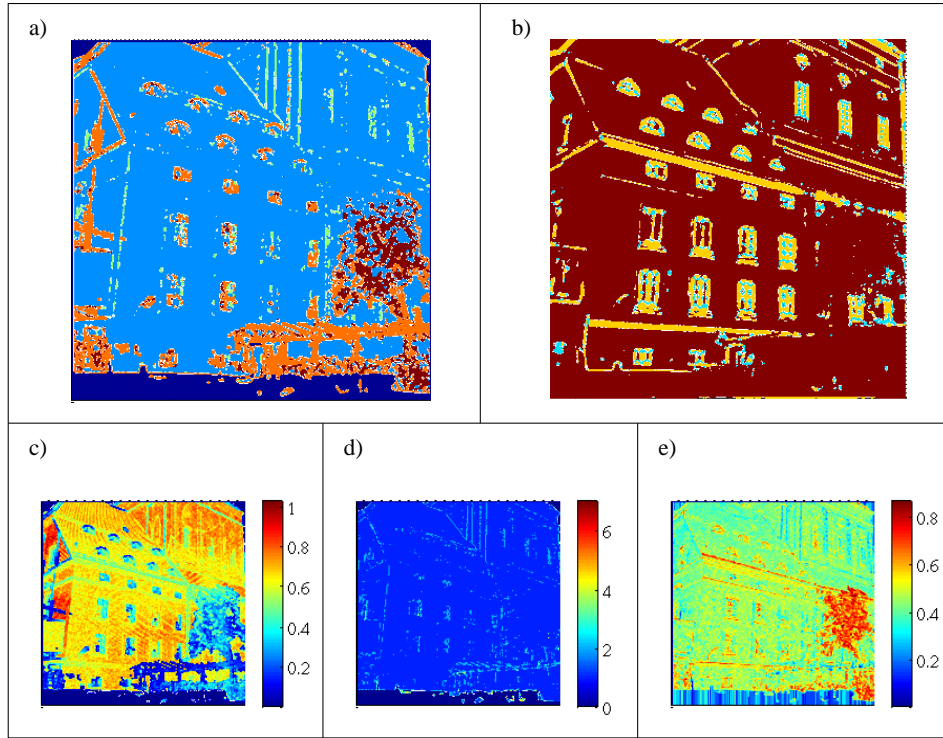
**Figure 7.** Results of the combined measures on artificial data. The camera and the range scanner are denoted by c. (a) Gap discontinuity tests. There are two planes which are separated by a distance  $d$  where  $d = 0, 0.01, 0.02, 0.03, 0.04$  meters. (b) The detected discontinuities. Dark blue marks the boundary points where the measures are not applicable. Blue and orange respectively correspond to detected continuities and gap discontinuities. (c) Orientation discontinuity tests. There are two planes which are connected but separated with an angle  $a$  where  $a = 180, 171, 153, 117, 90$  degrees. (d) The detected discontinuities. Dark blue marks the boundary points where the measures are not applicable. Blue and green respectively correspond to detected continuities and orientation discontinuities.

- continuous if  $\mu_{GD}(P) \geq T_g$  and  $\mu_{OD}(P) \leq 1$ .

For our analysis,  $N$ , where  $N \times N$  is the size of the patches is set to 10 pixels. Bigger values for  $N$  means larger support region for the measures, in which case different kinds of 3D discontinuities might interfere in the patch. On the other hand, using smaller values would make the measures very sensitive to noise. Other thresholds  $T_g$  and  $T_{ig}$  are respectively set to 0.4 and 0.6. These values are empirically determined by testing the measures over a large set of samples. Different values for these thresholds may result in wrong classifications of local 3D structures and may lead to different results than presented in this paper. Similarly, the number of bins,  $n$ , in  $H^n$  is empirically determined as 20.

Figure 7 shows the performance of the measures on two artificial scenes, one for gap discontinuity and one for orientation discontinuity for a set of depth and angle differences between planes. In the figure, the detected discontinuity type is shown for each pixel. We see that gap discontinuity can be detected reliable even if the gap difference is low. The sensitivity





**Figure 8.** The 3D and 2D information for one of the scenes shown in figure 5. Dark blue marks the points without range data. (a) 3D discontinuity. Blue: continuous surfaces, light blue: orientation discontinuities, orange: gap discontinuities and brown: irregular gap discontinuities. (b) Intrinsic Dimensionality. Homogeneous patches, edge-like and corner-like structures are encoded in colors brown, yellow and light blue, respectively. (c) Gap discontinuity measure  $\mu_{GD}$ . (d) Orientation discontinuity measure  $\mu_{OD}$ . (e) Irregular gap discontinuity measure  $\mu_{IGD}$ .

Dis. Type	$\mu_{GD}$	$\mu_{IGD}$	$\mu_{OD}$
<i>Continuity</i>	High value	Don't care	1
<i>Gap Dis.</i>	Low value	Low value	Don't care
<i>Irregular Gap Dis.</i>	Low value	High value	Don't care
<i>Orientation Dis.</i>	High value	Don't care	$> 1$

**Table 1.** The relation between the measurements and the types of the 3D discontinuities.

of the orientation discontinuity measure is around 160 degrees. However, the sensitivity of the measures would be different in real scenes due to the noise in the range data.

For a real example scene from figure 5, the detected discontinuities are shown in figure 8(a). We see that the underlying 3D structure of the scene is reflected in figure 8(a).

Note that this categorical combination of the measures appears to be against the motivation that has been provided for the classification of local 2D structures where we had advocated a continuous approach. There are two reasons: (1) With continuous 3D measures, the dimensionality of the results would be four (origin variance, line variance, a 3D measure and the normalized frequency of the signals), which is difficult to visualize and analyse. In



fact, the number of triangles that had to be shown in figure 9 would be 12, and it would be very difficult to interpret all the triangles together. (2) It has been argued by several studies [Huang et al., 2000, Yang and Purves, 2003] that range images are much simpler and less complex to analyze than 2D images. This suggests that it might be safer to have a categorical classification for range images.

#### 4. First-order Statistics: Analysis of the Relation Between Local 3D and 2D Structure

In this section, we analyze the relation between local 2D structures and local 3D structure; namely, the likelihood of observing a 3D structure given the corresponding 2D structure (*i.e.*,  $P(3D \text{ Structure} | 2D \text{ Structure})$ ).

##### 4.1. Results and Discussion

For each pixel of the scene (except where range data is not available), we computed the 3D discontinuity type and the intrinsic dimensionality. Figures 8(a) and (b) show the images where the 3D discontinuity and the intrinsic dimensionality of each pixel are marked with different colors.

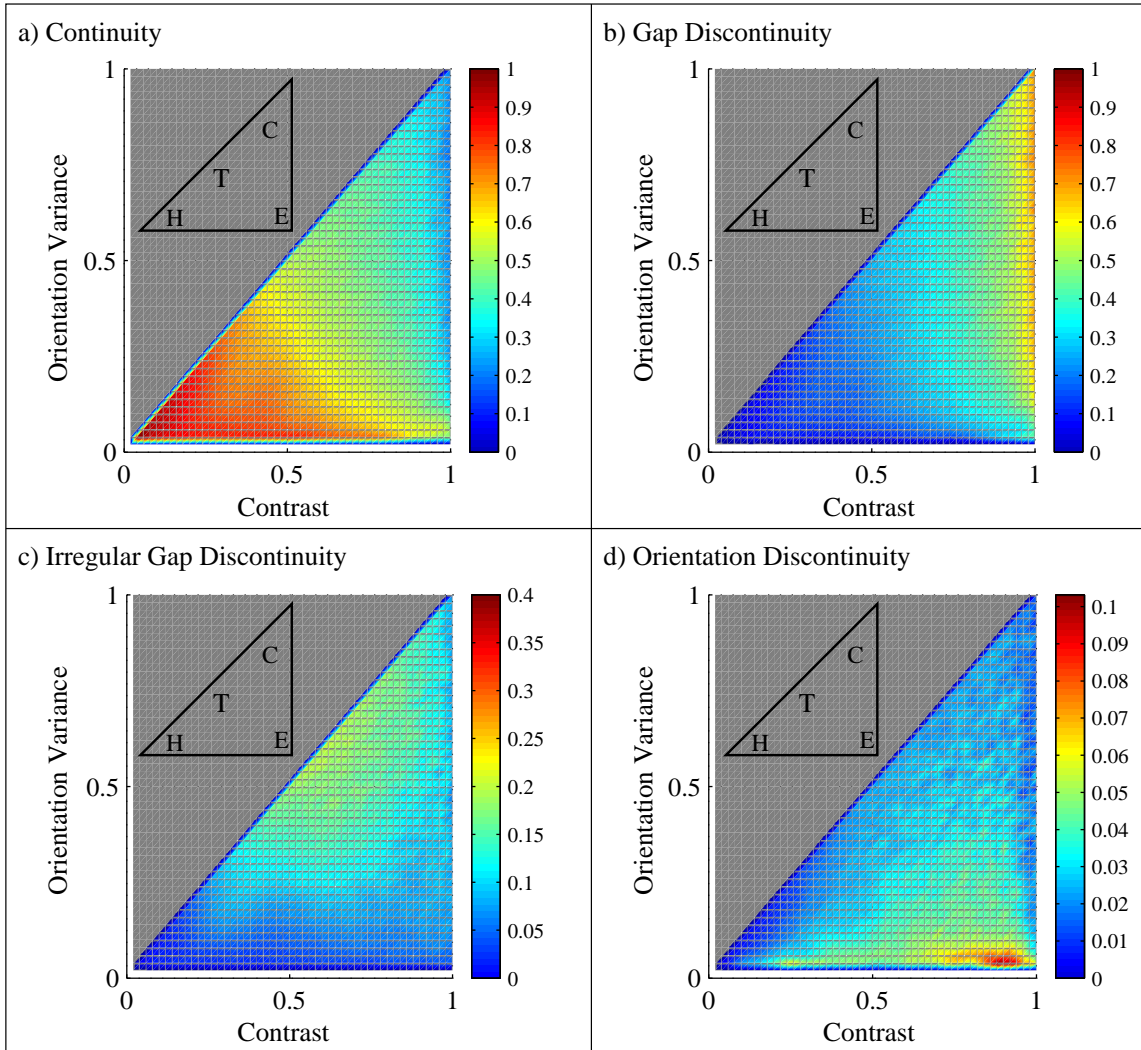
Having the 3D discontinuity type and the information about the local 2D structure of each point, we wanted to analyze what the likely underlying 3D structure is for a given local 2D structure; that is, the conditional likelihood  $P(3D \text{ Discontinuity} | 2D \text{ Structure})$ . Using the available 3D discontinuity type and the information about the local 2D structure, other measurements or correlations between the range data and the image data could also be computed in a further study.

$P(3D \text{ Discontinuity} | 2D \text{ Structure})$  is shown in figure 9. Note that the four triangles in figures 9(a), 9(b), 9(c) and 9(d) add up to one for all points of the triangle.

In figure 10, maximum likelihood estimates (MLE) of local 3D structures given local 2D structures are provided. Figure 10(a) shows the MLE from the distributions in figure 9. Due to high likelihoods, gap discontinuities and continuities are the most likely estimates given local 2D structures. Figure 10(b) shows the MLE from the *normalized* distributions: *i.e.*, each triangle in figure 9 is normalized within itself so that its maximum likelihood is 1. This way we can see the mostly likely *local 2D structures* for different local 3D structures.

- Figure 9(a) shows that homogeneous 2D structures are very likely to be formed by 3D continuities as the likelihood  $P(\text{Continuity} | 2D \text{ Structure})$  is very high (bigger than 0.85) for the area where homogeneous 2D structures exist (marked with H in figure 9(a)). This observation is confirmed in the MLE estimates of figure 10.

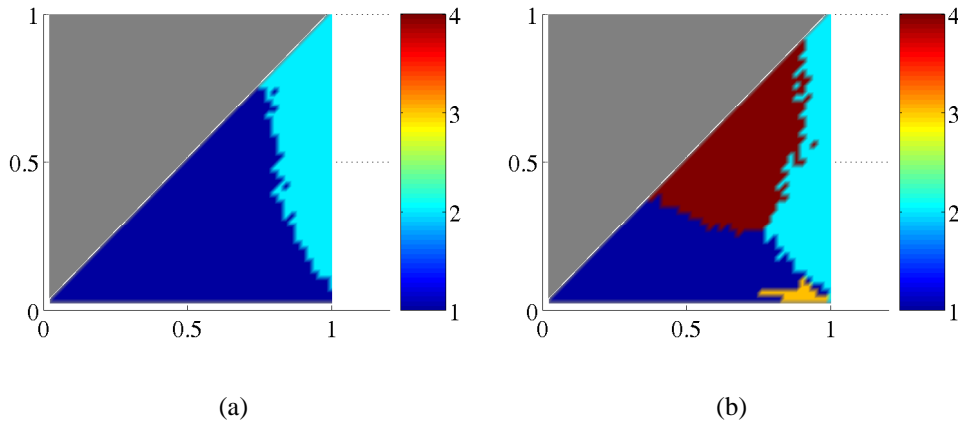
Many surface reconstruction studies make use of a basic assumption that there is a smooth surface between any two points in the 3D world, if there is no contrast difference between these points in the image. This assumption has been first called as 'no news is good news' in [Grimson, 1983]. Figure 9(a) quantifies 'no news is good news' and shows for which structures and to what extent it holds: In addition to the fact that no news is in fact good news, figure 9(a) shows that news, especially texture-like structures and



**Figure 9.**  $P(3D \text{ Discontinuity} \mid 2D \text{ Structure})$ . The schematic insets indicate the locations of the different types of 2D structures inside the triangle for easy reference (the letters C, E, H, T represent corner-like, edge-like, homogeneous and texture-like structures). (a)  $P(\text{Continuity} \mid 2D \text{ Structure})$ . (b)  $P(\text{Gap Discontinuity} \mid 2D \text{ Structure})$ . (c)  $P(\text{Irregular Gap Discontinuity} \mid 2D \text{ Structure})$ . (d)  $P(\text{Orientation Discontinuity} \mid 2D \text{ Structure})$ .

edge-like structures, can also be good news (see below). Homogeneous 2D structures cannot be used for depth extraction by correspondence-based methods, and only weak or no information from these structures is processed by the cortex. Unfortunately, the vast majority of local image structure is of this type (see, *e.g.*, [Kalkan et al., 2005]). On the other hand, homogeneous structures indicate 'no change' in depth which is the underlying assumption of interpolation algorithms.

- Edges are considered as important sources of information for object recognition and reliable correspondence finding. Approximately 10% of local 2D structures are of that type (see, *e.g.*, [Kalkan et al., 2005]). Figures 9(a), (b) and (d) together with the MLE estimates in figure 10 show that most of the edges are very likely to be formed by



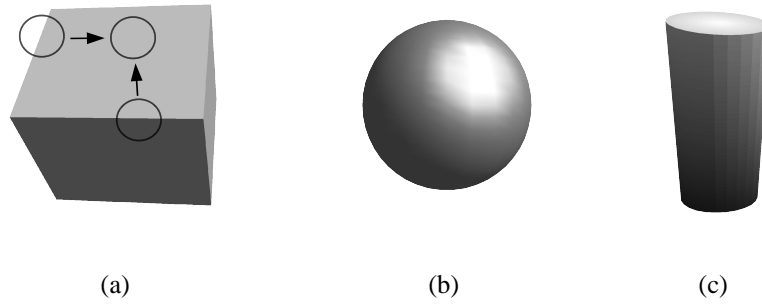
**Figure 10.** Maximum likelihood estimates of local 3D structures given local 2D structures. Numbers 1, 2, 3 and 4 represent continuity, gap discontinuity, orientation discontinuity and irregular gap discontinuity, respectively. **(a)** Raw maximum likelihood estimates. Note that the estimates are dominated by continuities and gap discontinuities. **(b)** Maximum likelihood estimates from normalized likelihood distributions: the triangles provided in figure 9 are normalized within themselves so that the maximum likelihood of  $P(X | \text{2D Structure})$  is 1 for  $X$  being continuity, gap discontinuity, irregular gap discontinuity and orientation discontinuity.

continuous surfaces or gap discontinuities. Looking at the decision areas for different local 2D structures shown in figure 2(d), we see that the edges formed by continuous surfaces are mostly low-contrast edges (figure 9(a)); *i.e.*, the origin variance is close to 0.5. Little percentage of the edges are formed by orientation discontinuities (figure 9(d)).

- Figures 9(a) and (b) show that well-defined corner-like structures are formed by either gap discontinuities or continuities.
- Figures 9(d) and 10 show that textures also are very likely to be formed by surface continuities and irregular gap discontinuities.

Finding correspondences becomes more difficult with the lack or repetitiveness of the local structure. The estimates of the correspondences at texture-like structures are naturally less reliable. In this sense, the likelihood that certain textures are formed by continuous surfaces (shown in figure 9(a)) can be used to model stereo matching functions that include interpolation as well as information about possible correspondences based on the local image information.

It is remarkable that local 2D structures mapping to different sub-regions in the triangle are formed by rather different 3D structures. This clearly indicates that these different 2D structures should be used in different ways for surface reconstruction.



**Figure 11.** Illustration of the relation between the depth of homogeneous 2D structures and the bounding edges. **(a)** In the case of the cube, the depth of homogeneous image area and the bounding edges are related. However, in the case of round surfaces, **(b)** the depth of homogeneous 2D structures may not be related to the depth of the bounding edges. **(c)** In the case of a cylinder, we see both cases of the relation as illustrated in (a) and (b).

## 5. Second-Order Statistics: Analysis of Co-planarity between 3D Edges and Continuous Patches

As already mentioned in section 1, it is not possible to extract depth at homogeneous 2D structures (in the rest of the paper, a homogeneous 2D structure that corresponds to a 3D continuity will be called a *mono*) using methods that make use of multiple views for 3D reconstruction. In this section, by making use of the ground truth range data, we investigate co-planarity relations between the depth at homogeneous 2D structures and the edges that bound them. This relation is illustrated for a few examples in figure 11.

For the analysis, we used the chromatic range data set that we also used for the first-order analysis in section 4. Samples from the dataset are displayed in figure 5.

In the following subsection, we explain how we analyze the relation. The results are presented and discussed in section 5.2.

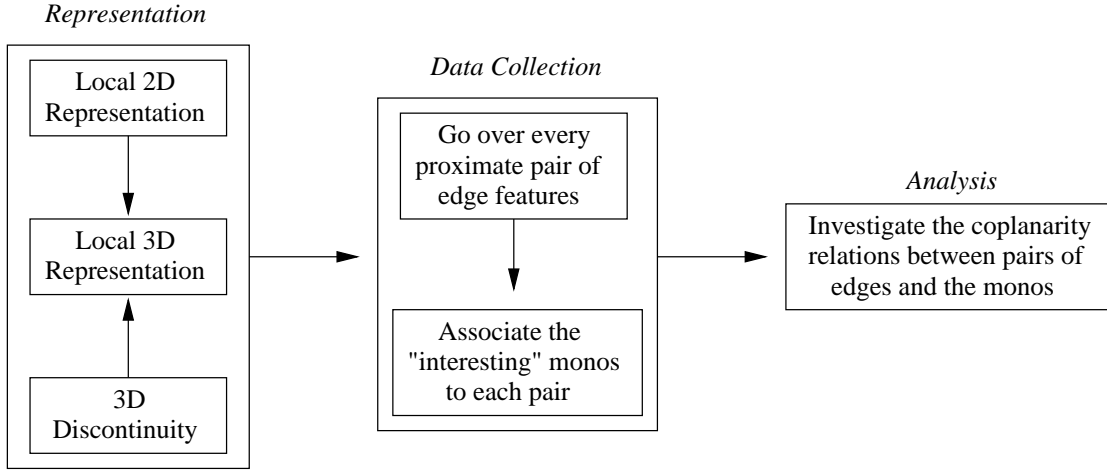
### 5.1. Methods

This subsection provides the procedural details of how the analysis is performed.

The analysis is performed in three stages: First, local 2D and 3D representations of the scene are extracted from the chromatic range data. Second, a data set is constructed out of each pair of edge features, associating the monos that are likely to be coplanar to those edges to them (see section 5.1.2 for what we mean by relevance). Third, the coplanarity between the monos and the edge features that they are associated to are investigated. An overview of the analysis process is sketched in figure 12, which roughly lists the steps involved.

#### 5.1.1. Representation

Using the 2D image and the associated 3D range data, a representation of the scene is created in terms of local compository 2D and 3D features denoted by  $\pi$ . In this process, first, 2D features are extracted from the image information, and at the locations of these 2D



**Figure 12.** Overview of the analysis process. First, local 2D and 3D representations of the scene are extracted from the chromatic range data. Second, a data set is constructed out of each pair of edge features, associating the monos that are likely to be coplanar (*i.e.*, "interesting") to them (see section 5.1.2 for what we mean by relevance). Third, the coplanarity between the monos and the edge features that they are associated to are investigated.

features, 3D features are computed. The complementary information from the 2D and 3D features are then merged at each valid position, where validity is only defined by having enough range data to extract a 3D representation.

For homogeneous and edge-like structures, different representations are needed due to different underlying structures. For this reason, we have two different definitions of  $\pi$  denoted respectively by  $\pi^e$  (for edge-like structures) and  $\pi^m$  (for monos) and formulated as:

$$\pi^m = (\mathbf{X}_{3D}, \mathbf{X}_{2D}, \mathbf{c}, \mathbf{p}), \quad (9)$$

$$\pi^e = (\mathbf{X}_{3D}, \mathbf{X}_{2D}, \phi_{2D}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{p}_1, \mathbf{p}_2), \quad (10)$$

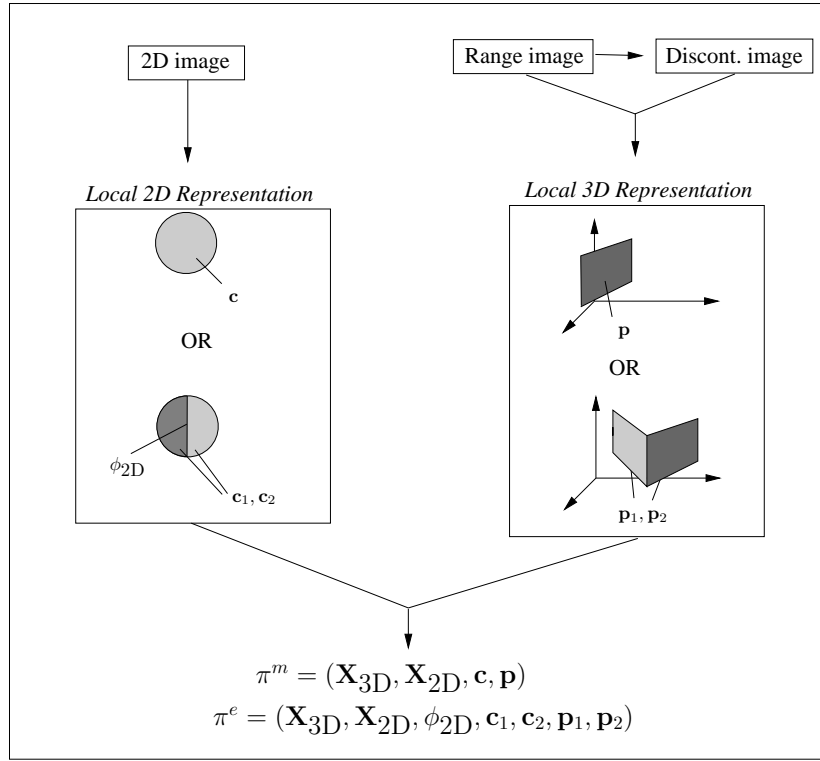
where  $\mathbf{X}_{3D}$  and  $\mathbf{X}_{2D}$  denote 3D and 2D positions of the 3D entity;  $\phi_{2D}$  is the 2D orientation of the 3D entity;  $\mathbf{c}_1$  and  $\mathbf{c}_2$  are the 2D color representation of the surfaces of the 3D entity;  $\mathbf{c}$  represents the color of  $\pi^m$ ;  $\mathbf{p}_1$  and  $\mathbf{p}_2$  are the planes that represent the surfaces that meet at the 3D entity; and  $\mathbf{p}$  represents the plane of  $\pi^m$  (see figure 13). Note that  $\pi^m$  does not have any 2D orientation information (because it is undefined for homogeneous structures), and  $\pi^e$  has two color and plane representations to the 'left' and 'right' of the edge.

The process of creating the representation of a scene is illustrated in figure 13.

In our analysis, the entities are regularly sampled from the 2D information. The sampling size is 10 pixels. See [Krüger et al., 2003, Krüger and Wörgötter, 2005] for details.

Extraction of the planar representation requires knowledge about the type of local 3D structure of the 3D entity (see figure 13). Namely, if the 3D entity is a continuous surface, then only one plane needs to be extracted; if the 3D entity is an orientation discontinuity, then there will be two planes for extraction; if the 3D entity is a gap discontinuity, then there will also be two planes for extraction.

In the case of a continuous surface, a single plane is fitted to the set of 3D points in the 3D entity in question. For orientation discontinuous 3D structures, extraction of the



**Figure 13.** Illustration of the representation of a 3D entity. From the 2D and 3D information, local 2D and 3D representation is extracted.

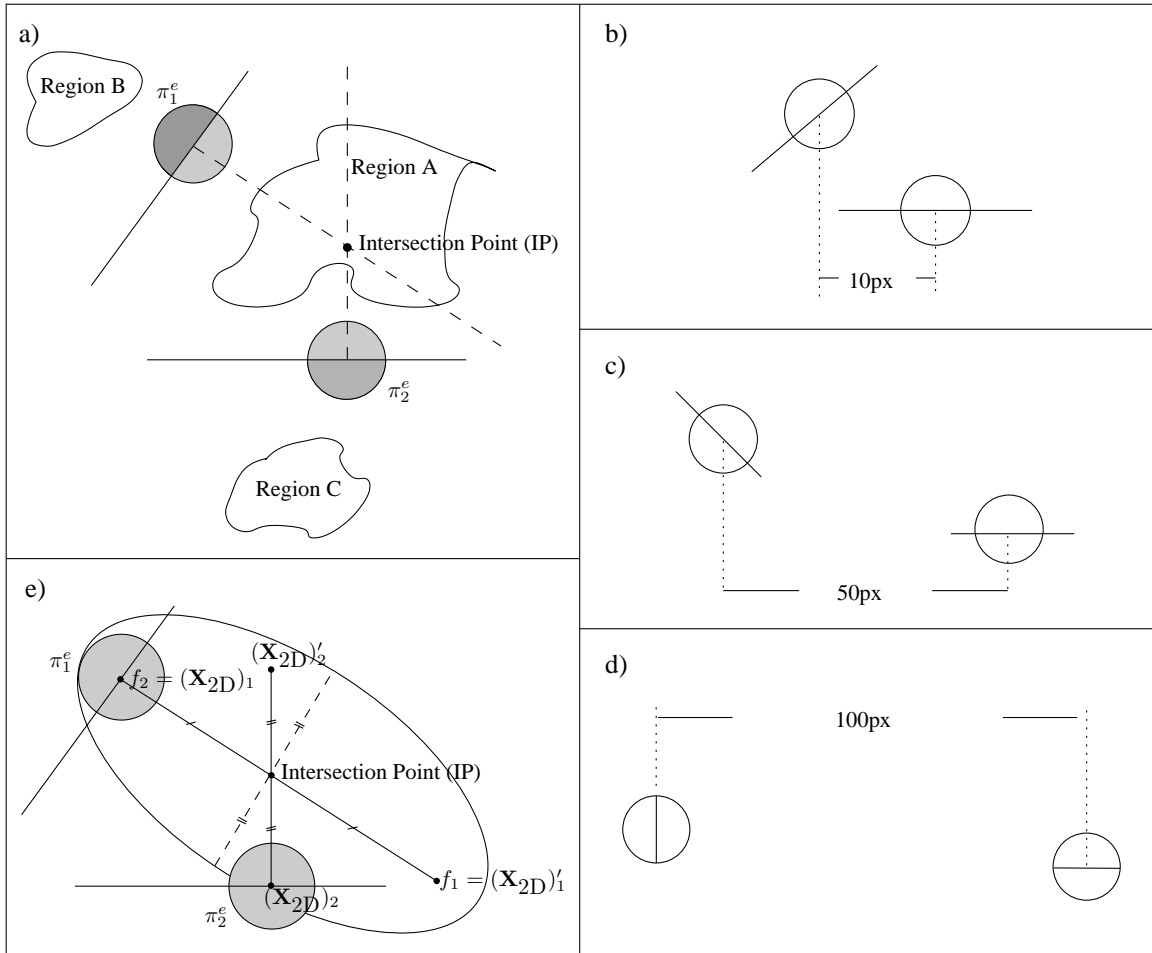
planar representation is not straight-forward. For these structures, our approach was to fit unit-planes<sup>–</sup> to the 3D points of the 3D entity and find the two clusters in these planes using k-means clustering of the 3D orientations of the small planes. Then, one plane is fitted for each of the two clusters, producing the bi-fold planar representation of the 3D entity.

Color representation is extracted in a similar way. If the image patch is a homogeneous structure, then the average color of the pixels in the patch is taken to be the color representation. If the image patch is edge-like, then it has two colors separated by the line which goes through the center of the image patch and which has the 2D orientation of the image patch. In this case, the averages of the colors of the different sides of the edge define the color representation in terms of  $c_1$  and  $c_2$ . If the image patch is corner-like, the color representation becomes undefined.

### 5.1.2. Collecting the Data Set

In our analysis, we form pairs out of  $\pi^e$ s that are close enough (see below), and for each pair, we check whether monos in the scene are coplanar to the elements of the pair or not. As there are plenty of monos in the scene, we only consider a subset of monos for each pair of  $\pi^e$  that we suspect to be relevant to the analysis because otherwise, the analysis becomes computationally intractable. The situation is illustrated in figure 14(a). In this figure, two  $\pi^e$  and three regions are shown; however, only one of these regions (*i.e.*, region A) is likely to

<sup>–</sup> By unit-planes, we mean planes that are fitted to the 3D points that are 1-pixel apart in the 2D image.



**Figure 14.** (a) Given a pair of edge features, coplanarity relation can be investigated for homogeneous image patches inside regions A, B and C. However, due to computational intractability reasons, this paper is concerned in making the analysis only in region A (see the text for more details). (b)-(d) A few different configurations of edge features that might be encountered in the analysis. The difficult part of the investigation is to make these different configurations comparable, which can be achieved by fitting a shape (like square, rectangle, circle, parallelogram, ellipse) to these configurations. (e) The ellipse, among the alternative shapes (*i.e.*, square, rectangle, circle, parallelogram) turns out to describe the different configurations shown in (b)-(d) better. For this reason, ellipse is for analyzing coplanarity relations in the rest of the paper. See the text for details on how the parameters of the ellipse are set.

have coplanar monos (*e.g.*, see figure 11(a)). This *assumption* is based on the observation of how objects are formed in the real world: objects have boundaries which consists of edge-like structures who bound surfaces, or image areas, of the object. The image area that is bounded by a pair of edge-like structures is likely to be the area that has the normals of both structures. For convex surfaces of the objects, the area that is bounded belongs to the object; however, in the case of concave surfaces, the area covered may also be from other objects, and the extent of the effect of this is part of the analysis.

Let  $\mathcal{P}$  denote the set of pairs of proximate  $\pi^e$ s whose normals intersect.  $\mathcal{P}$  can be defined

as:

$$\mathcal{P} = \left\{ (\pi_1^e, \pi_2^e) \mid \forall \pi_1^e, \pi_2^e, \pi_1^e \in \Omega(\pi_2^e), I(\perp(\pi_1^e), \perp(\pi_2^e)) \right\}, \quad (11)$$

where  $\Omega(\pi^e)$  is the N-pixel-2D-neighborhood of  $\pi^e$ ;  $\perp(\pi^e)$  is the 2D line orthogonal to the 2D orientation of  $\pi^e$ , i.e., the normal of  $\pi^e$ ; and,  $I(l_1, l_2)$  is true if the lines  $l_1$  and  $l_2$  intersect. We have taken N to be 100.

It turns out that there are a lot of different configurations possible for a pair of edge features based on relative position and orientation, which are illustrated for a few cases in figure 14(b)-(d). The difficult part of the investigation is to be able to compare these different configurations. One way to achieve this is to fit a shape to region A which can *normalize* the coplanarity relations by its size in order to make them comparable (see section 5.2 for more information).

The possible shapes would be square, rectangle, parallelogram, circle and ellipse. Among the alternatives, it turns out that an ellipse (1) is computationally cheap and (2) fits to different configurations of  $\pi_1$  and  $\pi_2$  under different orientations and distances *without* leaving region A much. Figure 14(e) demonstrates the ellipse generated by an example pair of edges in figure 14(a). The center of the ellipse is at the intersection of the normals of the edges, which we call *the intersection point* (IP) in the rest of the paper.

The parameters of an ellipse are composed of two focus points  $f_1, f_2$  and the minor axis  $b$ . In our analysis, the more distant 3D edge determines the foci of the ellipse (and, hence, the major axis), and the other 3D edge determines the length of the minor axis. Alternatively, the ellipse can be constructed by minimizing an energy functional which optimizes the area of the ellipse inside region A and going through the features  $\pi_1$  and  $\pi_2$ . However, for the sake of speed issues, the ellipse is constructed without optimization.

See appendix A.1 for details on how we determine the parameters of the ellipse.

For each pair of edges in  $\mathcal{P}$ , the region to analyze coplanarity is determined by intersecting the normals of the edges. Then, the monos inside the ellipse are associated to the pair of edges.

Note that a  $\pi^e$  has two planes that represent the underlying 3D structure. When  $\pi^e$ s become associated to monos, only one plane, the one that points into the ellipse, remains relevant. Let  $\pi^{se}$  denote the semi-representation of  $\pi^e$  which can be defined as:

$$\pi^{se} = (\mathbf{X}_{3D}, \mathbf{X}_{2D}, \mathbf{c}, \mathbf{p}). \quad (12)$$

Note that  $\pi^{se}$  is equivalent to the definition of  $\pi^m$  in equation 10.

Let  $\mathcal{T}$  denote the data set which stores  $\mathcal{P}$  and the associated monos which can be formulated as:

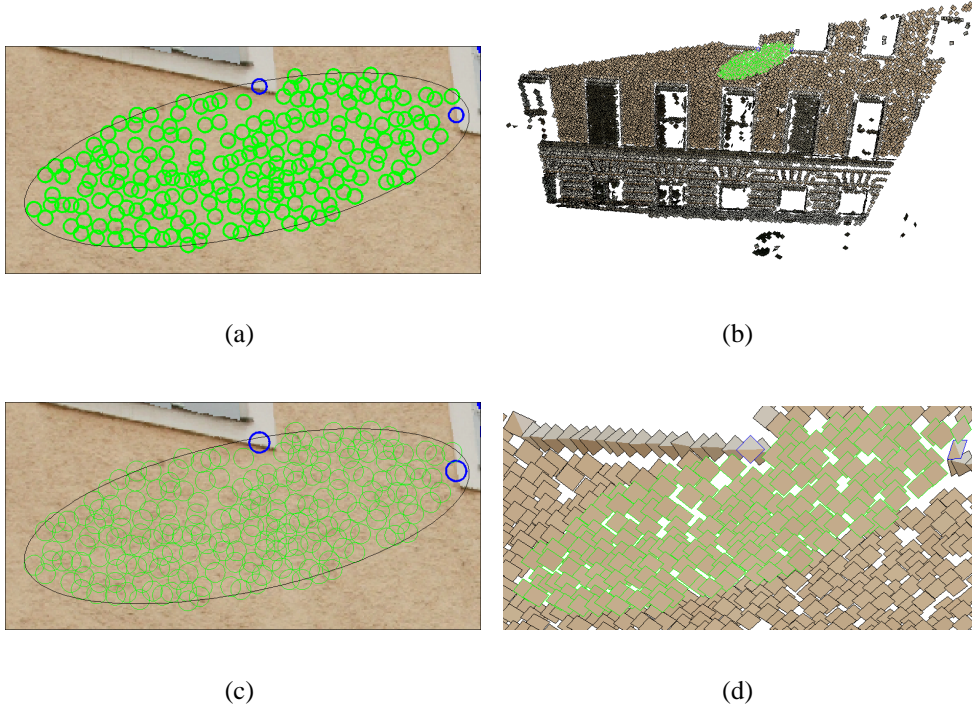
$$\mathcal{T} = \{ (\pi_1^{se}, \pi_2^{se}, \pi^m) \mid (\pi_1^e, \pi_2^e) \in \mathcal{P}, \pi^m \in \mathcal{S}^m, \pi^m \in E(\pi_1^e, \pi_2^e) \}, \quad (13)$$

where  $\mathcal{S}^m$  is the set of all  $\pi^m$ .

A pair of  $\pi^e$ s and the set of monos associated to them are illustrated in figure 15. The figure shows the edges and the monos (together with ellipse) in 2D and 3D.

<sup>o</sup> In other words, the Euclidean image distance between the structures should be less than N.





**Figure 15.** Illustration of a pair of  $\pi^e$  and the set of monos associated to them. **(a)** The input scene. A pair of edges (marked in blue) and the associated monos (marked in green) with an ellipse (drawn in black) around them shown on the input image. See (c) for a zoomed version. **(b)** The 3D representation of the scene in our 3D visualization software. This representation is created from the range data corresponding to (a) and is explained in the text. **(c)** The part of the input image from (a) where the edges, the monos and the ellipse are better visible. **(d)** A part of the 3D representation (from (b)) corresponding to the pair of edges and the monos in (c) is displayed in detail where the edges are shown with blue margins; the monos with the edges are shown in green (all monos are coplanar with the edges). The 3D entities are drawn in rectangles because of the high computational complexity for drawing circles.

### 5.1.3. Definition of coplanarity

Two entities are coplanar if they are on the same plane. Coplanarity of edge features and monos is equivalent to coplanarity of two planar patches: two planar patches  $A$  and  $B$  are coplanar if (1) they are parallel and (2) the planar distance between them is zero.

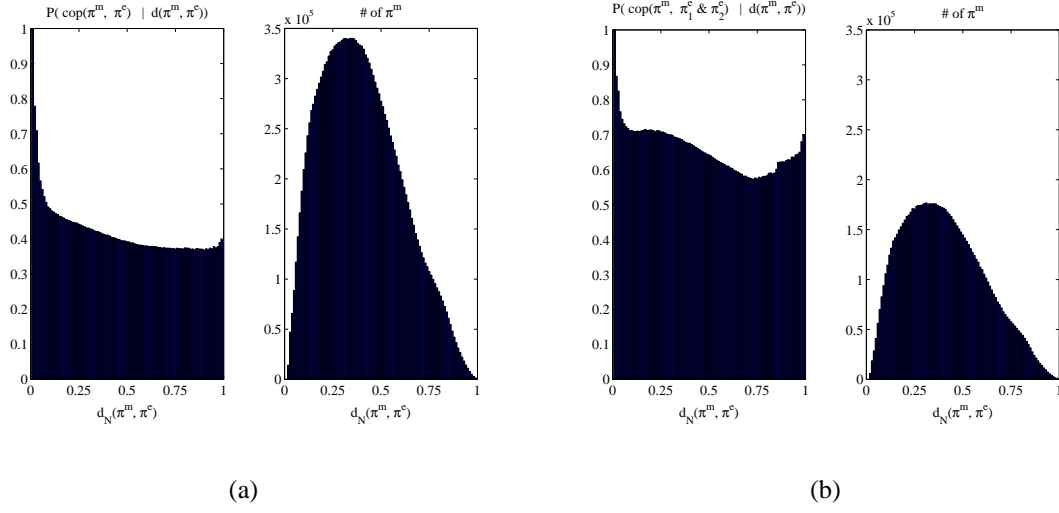
See appendix A.2 for more information.

## 5.2. Results and Discussions

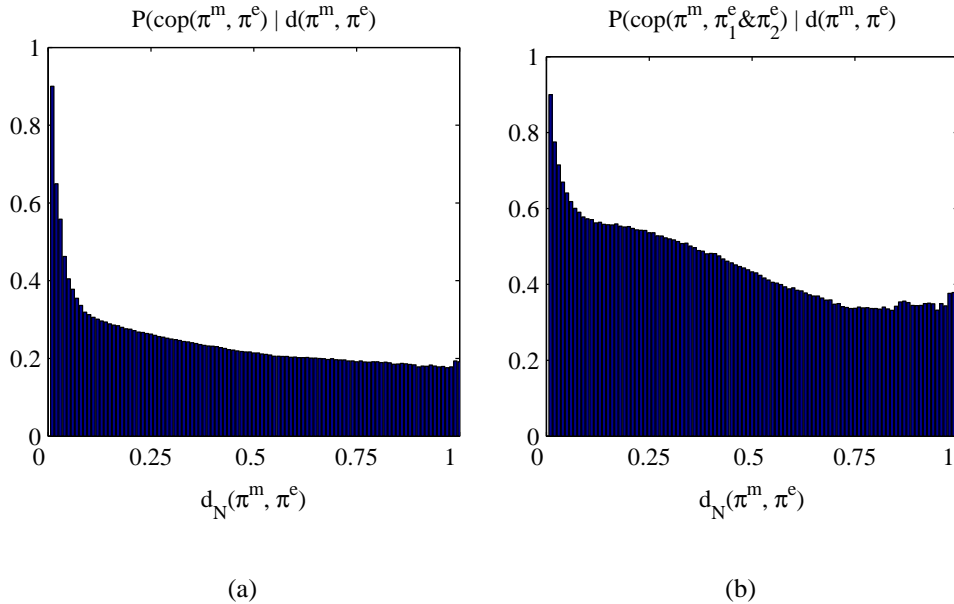
The data set  $\mathcal{T}$  defined in equation 13 consists of pairs of  $\pi_1^e$ ,  $\pi_2^e$  and the associated monos. Using this set, we compute the likelihood that a mono is coplanar with  $\pi_1^e$  and/or  $\pi_2^e$  against a distance measure.

The results of our analysis are shown in figures 16 and 18 and 19.

In figure 16(b), the likelihood of the coplanarity of a mono against the distance to  $\pi_1^e$  or  $\pi_2^e$  is shown. This likelihood can be denoted formally as  $P(\text{cop}(\pi^m, \pi_1^e \& \pi_2^e) \mid d_N(\pi^m, \pi^e))$

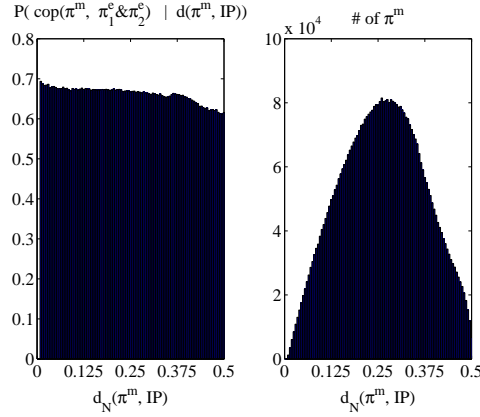


**Figure 16.** Likelihood distribution of coplanarity of monos. In each sub-figure, left-plot shows the likelihood distribution whereas right-plot shows the frequency distribution. (a) The likelihood of the coplanarity of a mono with  $\pi_1^e$  or  $\pi_2^e$  against the distance to  $\pi_1^e$  or  $\pi_2^e$ . This is the unconstrained case; *i.e.*, the case where there is no information about the coplanarity of  $\pi_1^e$  and  $\pi_2^e$ . (b) The likelihood of the coplanarity of a mono with  $\pi_1^e$  and  $\pi_2^e$  against the distance to  $\pi_1^e$  or  $\pi_2^e$ .

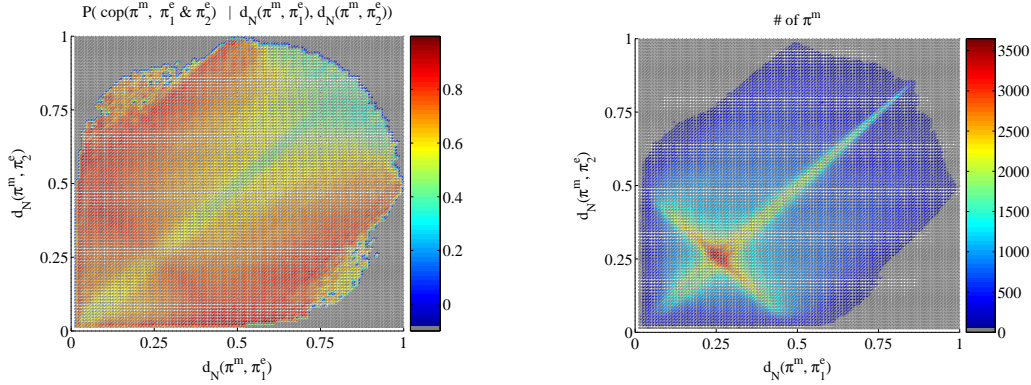


**Figure 17.** Likelihoods from figures 16(a) and 16(b) with a more *strict* coplanarity relation (namely, we set the thresholds  $T_p$  and  $T_d$  to 10 degrees and 0.2, respectively. See Appendix for more information about these thresholds). (a) Figure 16(a) with more strict coplanarity relation. (b) Figure 16(b) with more strict coplanarity relation.

where  $\text{cop}(\pi^m, \pi_1^e \& \pi_2^e)$  is defined as  $\text{cop}(\pi_1^e, \pi_2^e) \wedge \text{cop}(\pi^m, \pi^e)$ , and  $\pi^e$  is either  $\pi_1^e$  or  $\pi_2^e$ .



**Figure 18.** The likelihood of the coplanarity of a mono against the distance to  $IP$ . Left-plot shows the likelihood distribution whereas right-plot shows the frequency distribution.



**Figure 19.** The likelihood of the coplanarity of a mono against the distance to  $\pi_1^e$  and  $\pi_2^e$ . Left-plot shows the likelihood distribution whereas right-plot shows the frequency distribution.

The normalized distance measure<sup>‡</sup>  $d_N(\pi^m, \pi^e)$  is defined as:

$$d_N(\pi^m, \pi^e) = \frac{d(\pi^m, \pi^e)}{2\sqrt{d(\pi_1^e, IP)^2 + d(\pi_2^e, IP)^2}}, \quad (14)$$

where  $\pi^e$  is either  $\pi_1^e$  or  $\pi_2^e$ , and  $IP$  is the intersection point of  $\pi_1^e$  and  $\pi_2^e$ . We see in figure 16(b) that the likelihood decreases when a mono is more distant from an edge. However, when the distance measure gets closer to one, the likelihood increases again. This is because, when a mono gets away from either  $\pi_1^e$  or  $\pi_2^e$ , it gets closer to the other  $\pi^e$ .

In figure 16(a), we see the unconstrained case of figure 16(b); *i.e.*, the case where there is no information about the coplanarity of  $\pi_1^e$  and  $\pi_2^e$ ; namely, the likelihood  $P(\text{cop}(\pi^m, \pi^e) | d_N(\pi^m, \pi^e))$  where  $\pi^e$  is either  $\pi_1^e$  or  $\pi_2^e$ . The comparison with figure 16(b) shows that the existence of another edge in the neighborhood increases the likelihood of finding coplanar structures. As there is no other coplanar edge in the neighborhood, the likelihood does not increase when the distance is close to one (compare with figure 16(b)).

<sup>‡</sup> In the following plots, the distance means the Euclidean distance in the image domain.

It is intuitive to expect symmetries in figure 16. However, as (1) the roles of  $\pi_1^e$  and  $\pi_2^e$  in the ellipse are fixed, and (2) one  $\pi^e$  is guaranteed to be on the major axis, and the other  $\pi^e$  may or may not be on the minor axis, the symmetry is not observable in figure 16.

To see the effect of the coplanarity relation on the results, we reproduced figures 16(a) and 16(b) with a more *strict* coplanarity relation (namely, we set the thresholds  $T_p$  and  $T_d$  to 10 degrees and 0.2, respectively. See Appendix for more information about these thresholds). The results with more constrained coplanarity relation are shown in figure 17. Although the likelihood changes quantitatively, the figure shows the qualitative behaviours that have been observed with the standard thresholds. Moreover, we cross-checked the results for subsets of the original dataset (results not provided here) and confirmed the same qualitative results.

In figure 18, the likelihood of the coplanarity of a mono against the distance to  $IP$  (i.e.,  $P(\text{cop}(\pi^m, \pi_1^e \& \pi_2^e) \mid d_N(\pi^m, IP))$ ) is shown. We see in the figure that the likelihood shows a flat distribution against the distance to  $IP$ .

In figure 19, the likelihood of the coplanarity of a mono against the distance to  $\pi_1^e$  and  $\pi_2^e$  (i.e.,  $P(\text{cop}(\pi^m, \pi_1^e \& \pi_2^e) \mid d_N(\pi^m, \pi_1^e), d_N(\pi^m, \pi_2^e))$ ) is shown. We see that when  $\pi^m$  is close to  $\pi_1^e$  or  $\pi_2^e$ , it is more likely to be coplanar with  $\pi_1^e$  and  $\pi_2^e$  than when it is equidistant to both edges. The reason is that, when  $\pi^m$  moves away from an equidistant point, it becomes closer to the other edge, in which case the likelihood increases as shown in figure 16(b).

The results, especially figures 16(b) and 16(a) confirm the importance of the relation illustrated in figure 11(a).

## 6. Discussion

### 6.1. Summary of the findings

Section 4.1 analyzed the likelihood  $P(\text{3D Structure} \mid \text{2D Structure})$ . In this section, we confirm and quantify the assumptions used in several surface interpolation studies. Our main findings from this section are as follows:

- As expected, homogeneous 2D structures are formed by continuous surfaces.
- Surprisingly, considerable amount of edges and texture-like structures are likely to be formed by continuous surfaces too. However, we confirm the expectation that gap discontinuities and orientation discontinuities are likely to be the underlying 3D structure for edge-like structures. As for texture-like structures, they may also be formed by irregular gap discontinuities.
- Corner-like structures, on the other hand, are mainly formed by gap discontinuities.

In section 5.2, we investigated the predictability of depth at homogeneous 2D structures. We confirm the basic assumption that closer entities are very likely to be coplanar. Moreover, we provide results showing that this likelihood increases if there are more edge features in the neighborhood.

## 6.2. Interpretation of the findings

Existing psychophysical experiments (see, *e.g.*, [Anderson et al., 2002, Collett, 1985]), computational theories (see, *e.g.*, [Barrow and Tenenbaum, 1981, Grimson, 1982, Terzopoulos, 1988]) and the observation that humans can perceive depth at weakly textured areas suggest that in the human visual system, *an interpolation process* is realized that, starting with the local analysis of edges, corners and textures, computes depth also in areas where correspondences cannot easily be found.

This paper was concerned with the analysis of the statistics that might be involved in such an interpolation process, by making use of chromatic range data.

In the first part (section 4), we analyzed which local 2D structures suggest a depth interpolation process. Using natural images, we showed that homogeneous 2D structures correspond to continuous surfaces, as suggested and utilized by some computational theories of surface interpolation (see, *e.g.*, [Grimson, 1983]). On the other hand, a considerable proportion of edge-like structures lie on continuous surfaces (see figure 9(a)); *i.e.*, a contrast difference does not necessarily mean a depth discontinuity. This suggests that interpreting edges in combination with neighboring corners or edges is important for understanding the underlying 3D structure [Barrow and Tenenbaum, 1981].

The results from section 4 are useful in several contexts:

- Depth interpolation studies assume that homogeneous image regions are part of the same surface. Such studies can be extended with the statistics provided here as priors in a Bayesian framework. This extension would allow making use of the continuous surfaces that a contrast difference (caused by textures or edge-like structures) might correspond to.

Acquiring range data from a scene is a time-consuming task compared to image acquisition, which lasts on the order of seconds even for high resolutions. In [Torres-Mendez and Dudek, 2006], for mobile robot environment modeling, instead of making a full-scan of the whole scene, only partial range scan is performed due to time constraints. This partial range data is completed by using a Markov Random Field which is trained from a pair of complete range and the corresponding image data. In [Torres-Mendez and Dudek, 2006], the partial range data is produced in a regular way; *i.e.*, every  $n$ th scan-column is neglected. This assumption, however, may introduce aliasing in the 3D data acquired from natural images using depth cues, and therefore, their method may not be applicable. Nevertheless, it could possibly be improved by utilizing the priors introduced in this paper.

- Automated registration of range and color images of a scene is crucial for several purposes like extracting 3D models of real objects. Methods that align edges extracted from the intensity image with the range data already exist (see, *e.g.*, [Laycock and Day, 2006]). These methods can be extended with the results presented in this paper in a way that not only edges but also other 2D structures are used for alignment. Such an extension also allows a probabilistic framework by utilizing the likelihood  $P(3D \text{ Structure} \mid 2D \text{ Structure})$ . Moreover, making use of local 3D structure

types that are introduced in this paper can be more robust than just a gap discontinuity detection.

Such an extension is possible by maximizing the following energy function:

$$E(R, T) = \int_{u,v} P(\text{3D Structure at } (u, v) \mid \text{2D Structure at } (u, v)) du dv, \quad (15)$$

where  $R$  and  $T$  are translation and rotation of the range data in 3D space.

In the second part (section 5), we analyzed whether depth at homogeneous 2D structures is related to the depth of edge-like structures in the neighborhood. Such an analysis is important for understanding the possible mechanisms that could underlie depth interpolation processes. Our findings show that an edge feature provides significant evidence for making depth prediction at a homogeneous image patch that is in the neighborhood. Moreover, the existence of a second edge feature in its neighborhood which is not collinear with the first edge feature increases the likelihood of the prediction.

Using second order relations and higher order features for representing the 2D image and 3D range data, we produce confirming results that the range images are simpler to analyze compared to 2D images (see, [Huang et al., 2000, Yang and Purves, 2003]).

By extracting a more complex representation than existing range-data analysis studies, we could point to the intrinsic properties of the 3D world and its relation to the image data. This analysis is important because (1) it may be that the human visual system is adapted to the statistics of the environment [Brunswik and Kamiya, 1953, Knill and Richards, 1996, Krueger, 1998, Olshausen and Field, 1996, Purves and Lotto, 2002, Rao et al., 2002], and (2) it may be used in several computer vision applications (for example, depth estimation) in a similar way as in [Elder and Goldberg, 2002, Elder et al., 2003, Pugeault et al., 2004, Zhu, 1999].

In our current work, the likelihood distributions are being used for estimating the 3D depth at homogeneous 2D structures from the depth of bounding edge-like structures.

### 6.3. Limitations of the current work

The first limitation is due to the type of scenes that have been used; *i.e.*, scenes of man-made environments which also included trees. Alternative scenes could include pure forest scenes or scenes taken from an environment with totally round objects. However, we believe that our dataset captures the general properties of the scenes that a human being encounters in daily life.

Different scenes might produce quantitatively different but qualitatively similar results. For example, forest scenes would produce much more irregular gap discontinuities than the current scenes; however, our conclusions regarding the link between textures and irregular gap discontinuities would still hold. Moreover, coplanarity relations would be harder to predict for such scenes since (depending on the scale) surface continuities are harder to find; however, on a bigger scale, some forest scenes are likely to produce the same qualitative results presented in this paper because of piecewise planar leaves which are separated by gap discontinuities.

It should be noted that acquisition of range data with color images is very hard for forest scenes since the color image of the scene is taken after the scene is scanned with the scanner. During this period, the leaves and the trees may move (due to wind etc.), making the range and the color data inconsistent. In office environments, a similar problem arises: due to lateral separation between the digital camera and range scanner, there is the parallax problem, which again produces inconsistent range-color association. For an office environment, a small-scale range scanner needs to be used.

The statistics presented in this paper can be extended by analyzing forest scenes, office scenes etc. independently. The comparison of such independent analyses should provide more insights into the relations that this paper have investigated but we believe that the qualitative conclusions of this paper would still hold.

It would be interesting to see the results presented in the paper by changing the measure for surface continuity so that it can separate planar and curved surfaces. We believe that such a change would effect only the second part of the paper.

## 7. Acknowledgements

We would like to thank RIEGL UK Ltd. for providing us with 3D range data, and Nicolas Pugeault for his comments on the text. This work is supported by the European-funded DRIVSCO project, and an extension of two conference publications of the authors: [Kalkan et al., 2006, Kalkan et al., 2007].

## Appendix

### A.1. Parameters of an ellipse

Let us denote the position of two 3D edges  $\pi_1^e, \pi_2^e$  by  $(\mathbf{X}_{2D})_1$  and  $(\mathbf{X}_{2D})_2$  respectively. The vectors between the 3D edges and IP (let us call  $l_1$  and  $l_2$ ) can be defined as:

$$\begin{aligned} l_1 &= ((\mathbf{X}_{2D})_1 - IP), \\ l_2 &= ((\mathbf{X}_{2D})_2 - IP). \end{aligned} \tag{16}$$

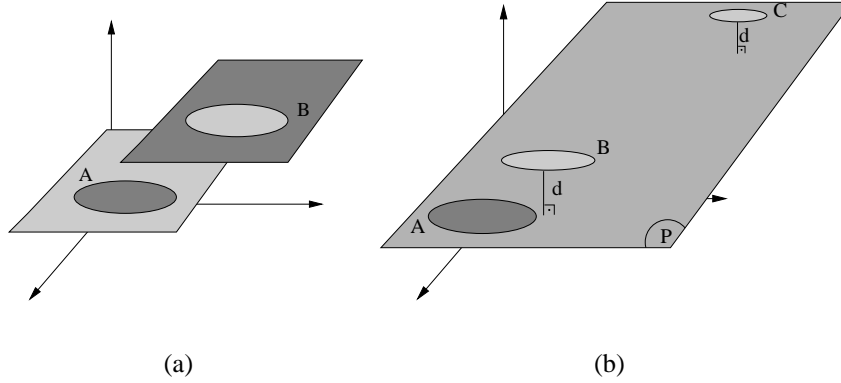
Having defined  $l_1$  and  $l_2$ , the ellipse  $E(\pi_1^e, \pi_2^e)$  is as follows:

$$E(\pi_1^e, \pi_2^e) = \begin{cases} f_1 = (\mathbf{X}_{2D})_1, f_2 = (\mathbf{X}_{2D})'_1, b = |l_2| & \text{if } |l_1| > |l_2|, \\ f_1 = (\mathbf{X}_{2D})_2, f_2 = (\mathbf{X}_{2D})'_2, b = |l_1| & \text{otherwise.} \end{cases} \tag{17}$$

where  $(\mathbf{X}_{2D})'$  is symmetrical with  $\mathbf{X}_{2D}$  around the intersection point and on the line defined by  $\mathbf{X}_{2D}$  and  $IP$  (as shown in figure 14(e)).

### A.2. Definition of coplanarity

Let  $\pi^s$  denote either a semi-edge  $\pi^{se}$  or a mono  $\pi^m$ . Two  $\pi^s$  are coplanar iff they are on the same plane. When it comes to measuring coplanarity, two criteria need to be tested:



**Figure 20.** Criteria for coplanarity of two planes. (a) According to the angular-difference criterion of coplanarity, entities A and B will be measured as coplanar although they are on different planes. In (b), P is the plane defined by entity A. According to the distance-based coplanarity definition, entities B and C have the same measure of coplanarity. However, entity C which is more distant to entity A should have a higher measure of coplanarity than entity B although they have the same distance to plane P (see the text).

- (i) Angular criterion: For two  $\pi^s$  to be coplanar, the angular difference between the orientation of the planes that represent them should be less than a threshold. A situation is illustrated in figure 20(a) where angular criterion holds but the planes are not coplanar.
- (ii) Distance-based criterion: For two  $\pi^s$  to be coplanar, the distance between the center of the first  $\pi^s$  and the plane defined by the other  $\pi^s$  should be less than a threshold. In figure 20(b), B and C are at the same distance to the plane P which is the plane defined by the planar patch A. However, C is more distant to the center of A than B, and in this paper, we treat that C is more coplanar to A than B is to A. The reason for this can be clarified with an example: Assume that A, B and C are all parallel, and that the *planar* and the Euclidean distances between A and B are both  $D$  units, and between A and C are respectively  $D$  and  $n \times D$ . It is straightforward to see that although B and C have the same planar distances to A, for  $n \gg 1$ , C should have a higher coplanarity measure.

It is sufficient to combine these two criteria as follows:

$$\begin{aligned} \text{cop}(\pi_1^s, \pi_2^s) &= \alpha(\mathbf{p}_1^{\pi_1^s}, \mathbf{p}_2^{\pi_2^s}) < T_p \text{ AND} \\ d(\mathbf{p}_1^{\pi_1^s}, \pi_2^s) / d(\pi_1^s, \pi_2^s) &< T_d, \end{aligned} \quad (18)$$

where  $\mathbf{p}^{\pi^s}$  is the plane associated to  $\pi^s$ ;  $\alpha(\mathbf{p}_1, \mathbf{p}_2)$  is the angle between the orientations of  $\mathbf{p}_1$  and  $\mathbf{p}_2$ ; and,  $d(\cdot, \cdot)$  is the Euclidean distance between two entities.

In our analysis, we have empirically chosen  $T_p$  and  $T_d$  as 20 degrees and 0.5, respectively. Again, like the parameters set in section 3.1.4, these values are determined by testing the coplanarity measure over different samples.  $T_p$  is the limit for angular separation between two planar patches. Bigger values would relax the coplanarity measure, and vice versa.  $T_d$  restricts the distances between the patches; in analogy to  $T_p$ ,  $T_d$  can be used to relax the coplanarity measure. As shown in figure 17 for a stricter coplanarity definition (with  $T_p$  and



$T_d$  set to 10 degrees and 0.2), different values for these thresholds would quantitatively but not qualitatively change the results presented in section 5.

## References

- [Anderson et al., 2002] Anderson, B. L., Singh, M., and Fleming, R. W. (March 2002). The interpolation of object and surface structure. *Cognitive Psychology*, 44:148–190(43).
- [Barrow and Tenenbaum, 1981] Barrow, H. G. and Tenenbaum, J. M. (1981). Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116.
- [Bolle and Vemuri, 1991] Bolle, R. M. and Vemuri, B. C. (1991). On three-dimensional surface reconstruction methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):1–13.
- [Bruce et al., 2003] Bruce, V., Green, P. R., and Georgeson, M. A. (2003). *Visual Perception: Physiology, Psychology and Ecology*. Psychology Press, 4th edition.
- [Brunswik and Kamiya, 1953] Brunswik, E. and Kamiya, J. (1953). Ecological cue–validity of ‘proximity’ and of other Gestalt factors. *American Journal of Psychologie*, LXVI:20–32.
- [Collett, 1985] Collett, T. S. (1985). Extrapolating and Interpolating Surfaces in Depth. *Royal Society of London Proceedings Series B*, 224:43–56.
- [Coxeter, 1969] Coxeter, H. (1969). *Introduction to Geometry (2nd ed.)*. Wiley & Sons.
- [Elder and Goldberg, 2002] Elder, H. and Goldberg, R. (2002). Ecological statistics of gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2(4):324–353.
- [Elder et al., 2003] Elder, J. H., Krupnik, A., and Johnston, L. A. (2003). Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(25):1–14.
- [Felsberg and Krüger, 2003] Felsberg, M. and Krüger, N. (2003). A probabilistic definition of intrinsic dimensionality for images. *Pattern Recognition, 24th DAGM Symposium*.
- [Field et al., 1993] Field, D. J., Hayes, A., and Hess, R. F. (1993). Contour integration by the human visual system: evidence for a local “association field”. *Vision Research*, 33(2):173–193.
- [Gallant et al., 1994] Gallant, J. L., Essen, D. C. V., and Nothdurft, H. C. (1994). *Early Vision and Beyond*, chapter : Two-dimensional and three-dimensional texture processing in visual cortex of the macaque monkey, pages 89–98. MA: MIT Press.
- [Grimson, 1982] Grimson, W. E. L. (1982). A Computational Theory of Visual Surface Interpolation. *Royal Society of London Philosophical Transactions Series B*, 298:395–427.
- [Grimson, 1983] Grimson, W. E. L. (1983). Surface consistency constraints in vision. *Computer Vision, Graphics and Image Processing*, 24(1):28–51.
- [Guzman, 1968] Guzman, A. (1968). Decomposition of a visual scene into three-dimensional bodies. *AFIPS Fall Joint Conference Proceedings*, 33:291–304.
- [Hoover et al., 1996] Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A., and Fisher, R. B. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689.
- [Howe and Purves, 2002] Howe, C. Q. and Purves, D. (2002). Range image statistics can explain the anomalous perception of length. *PNAS*, 99(20):13184–13188.
- [Howe and Purves, 2004] Howe, C. Q. and Purves, D. (2004). Size contrast and assimilation explained by the statistics of natural scene geometry. *Journal of Cognitive Neuroscience*, 16(1):90–102.
- [Huang et al., 2000] Huang, J., Lee, A. B., and Mumford, D. (2000). Statistics of range images. *CVPR*, 1(1):1324–1331.
- [Hubel and Wiesel, 1969] Hubel, D. and Wiesel, T. (1969). Anatomical demonstration of columns in the monkey striate cortex. *Nature*, 221:747–750.
- [Kalkan et al., 2005] Kalkan, S., Calow, D., Wörgötter, F., Lappe, M., and Krüger, N. (2005). Local image structures and optic flow estimation. *Network: Computation in Neural Systems*, 16(4):341–356.

- [Kalkan et al., 2006] Kalkan, S., Wörgötter, F., and Krüger, N. (2006). Statistical analysis of local 3d structure in 2d images. *CVPR*, 1:1114–1121.
- [Kalkan et al., 2007] Kalkan, S., Wörgötter, F., and Krüger, N. (2007). Statistical analysis of second-order relations of 3d structures. *Int. Conference on Computer Vision Theory and Applications (VISAPP)*.
- [Kellman and Arterberry, 1998] Kellman, P. and Arterberry, M., editors (1998). *The Cradle of Knowledge*. MIT-Press.
- [Knill and Richards, 1996] Knill, D. C. and Richards, W., editors (1996). *Perception as bayesian inference*. Cambridge: Cambridge University Press.
- [Koenderink and Dorn, 1982] Koenderink, J. and Dorn, A. (1982). The shape of smooth objects and the way contours end. *Perception*, 11:129–173.
- [Krueger, 1998] Krueger, N. (1998). Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129.
- [Krüger and Felsberg, 2003] Krüger, N. and Felsberg, M. (2003). A continuous formulation of intrinsic dimension. *Proceedings of the British Machine Vision Conference*.
- [Krüger et al., 2003] Krüger, N., Lappe, M., and Wörgötter, F. (2003). Biologically motivated multi-modal processing of visual primitives. *Proc. the AISB 2003 Symposium on Biologically inspired Machine Vision, Theory and Application, Wales*, pages 53–59.
- [Krüger and Wörgötter, 2005] Krüger, N. and Wörgötter, F. (2005). Multi-modal primitives as functional models of hyper-columns and their use for contextual integration. *Proc. 1st Int. Symposium on Brain, Vision and Artificial Intelligence, Naples, Italy, Lecture Notes in Computer Science, Springer, LNCS 3704*, pages 157–166.
- [Laycock and Day, 2006] Laycock, R. G. and Day, A. M. (2006). Image registration in a coarse three-dimensional virtual environment. *Computer Graphics Forum*, 25(1):69–82.
- [Lee et al., 1998] Lee, T. S., Mumford, D., Romero, R., and Lamme, V. A. F. (1998). The role of the primary visual cortex in higher level vision. *Vision Research*, 38:2429–2454.
- [Malik, 1987] Malik, J. (1987). Interpreting line drawings of curved objects. *International Journal of Computer Vision*, 1:73–103.
- [Marr, 1982] Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. Freeman.
- [Olshausen and Field, 1996] Olshausen, B. and Field, D. (1996). Natural image statistics and efficient coding. *Network*, 7:333–339.
- [Potetz and Lee, 2003] Potetz, B. and Lee, T. S. (2003). Statistical correlations between two-dimensional images and three-dimensional structures in natural scenes. *Journal of the Optical Society of America*, 20(7):1292–1303.
- [Pugeault et al., 2004] Pugeault, N., Krüger, N., and Wörgötter, F. (2004). A non-local stereo similarity based on collinear groups. *Proceedings of the Fourth International ICSC Symposium on Engineering of Intelligent Systems*.
- [Purves and Lotto, 2002] Purves, D. and Lotto, B., editors (2002). *Why we see what we do: an empirical theory of vision*. Sunderland, MA: Sinauer Associates.
- [Rao et al., 2002] Rao, R. P. N., Olshausen, B. A., and Lewicki, M. S., editors (2002). *Probabilistic models of the brain*. MA: MIT Press.
- [Rubin, 2001] Rubin, N. (2001). The role of junctions in surface completion and contour matching. *Perception*, 30:339–366.
- [Serenio et al., 2002] Sereno, M. E., Trinath, T., Augath, M., and Logothetis, N. K. (2002). Three-dimensional shape representation in monkey cortex. *Neuron*, 33(4):635–652.
- [Shirai, 1987] Shirai, Y. (1987). *Three-dimensional computer vision*. Springer-Verlag New York, Inc.
- [Simoncelli, 2003] Simoncelli, E. P. (2003). Vision and the statistics of the visual environment. *Current Opinion in Neurobiology*, 13(2):144–149.
- [Terzopoulos, 1988] Terzopoulos, D. (1988). The computation of visible-surface representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4):417–438.
- [Torres-Mendez and Dudek, 2006] Torres-Mendez, L. A. and Dudek, G. (2006). Statistics of visual and partial

- depth data for mobile robot environment modeling. *Mexican International Conference on Artificial Intelligence (MICAI)*.
- [Tuceryan and Jain, 1998] Tuceryan, M. and Jain, N. K. (1998). Texture analysis. *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, pages 207–248.
- [Yang and Purves, 2003] Yang, Z. and Purves, D. (2003). Image/source statistics of surfaces in natural scenes. *Network: Computation in Neural Systems*, 14:371–390.
- [Zetsche and Barth, 1990] Zetsche, C. and Barth, E. (1990). Fundamental limits of linear filters in the visual processing of two dimensional signals. *Vision Research*, 30(7):1111–1117.
- [Zhu, 1999] Zhu, S. C. (1999). Embedding gestalt laws in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1170–1187.

**Appendix E**  
***Perceptual Operations and Relations between 2D or 3D Visual Entities***  
S. Kalkan, N. Pugeault and N. Krüger  
( *Technical Report 2007 – 3* )

Robotics Group  
The Maersk Mc-Kinney Moller Institute  
University of Southern Denmark

---

Technical Report no. 2007 – 3

---

**Perceptual Operations and Relations between  
2D or 3D Visual Entities**

Sinan Kalkan, Nicolas Pugeault, Mogens Christiansen, Norbert Krüger

January 22, 2007

Title	Perceptual Operations and Relations between 2D or 3D Visual Entities
	Copyright © 2007 Sinan Kalkan, Nicolas Pugeault, Mogens Christiansen, Norbert Krüger. All rights reserved.
Author(s)	Sinan Kalkan, Nicolas Pugeault, Mogens Christiansen, Norbert Krüger
Publication History	

## Abstract

In this paper, we present a set of perceptual relations, namely, co-colority, co-planarity, collinearity and symmetry that are defined between multi-modal visual features that we call *primitives*.

## 1 Introduction

According to Marr’s paradigm [29], vision involves extraction of meaningful representations from input images, starting at the pixel level and building up its interpretation more or less in the following order: local filters, extraction of important features, the  $2\frac{1}{2}$ -D sketch and the 3-D sketch.

There is psychophysical evidence and evidence from the statistical properties of natural images that the human visual system utilizes a set of visual-entity-combining processes, called *perceptual organization* in the literature, for forming bigger, sparser and more complete interpretations of the scene (see, *e.g.*, [18, 19, 35]). Such processes include (i) extraction of the boundary of the objects in the image from the set of unconnected edge pixels or features [3, 8, 10, 21, 27, 31, 39] utilizing Gestalt laws of grouping, and (ii) interpolation and extrapolation of unconnected sparse 3D entities for forming more complete 3D surfaces (see, *e.g.*, [13]) utilizing the relations between the 3D entities. Gestalt principles include collinearity, proximity, common fate and similarity whereas inference of 3D surfaces from a set of 3D entities include relations like coplanarity, collinearity, co-colority etc. These are essentially second order and higher order relations of local features. In [26], we have introduced a specific form of a local descriptor that we call a ‘multi-modal primitive’ (see section 2) and which can be seen as a functional abstraction of a hypercolumn (see [24]). We distinguish between 2D primitives describing local image information and 3D primitives covering local 3D scene information in a condensed symbolic way.

These primitives serve as a basis for an early cognitive vision system [23, 26, 33] in which operations and relations on these primitives realizing perceptual grouping principles are used in different contexts (see [26] for applications). We have utilized these relations for different problems including stereo [34], RBM [32], estimation of initial grasping reflexes from stereo [5], estimation of depth at homogeneous image structures [16], and analysis of second-order relations between 3D features [17].

In this paper, we present the set of 2D and 3D relations defined upon the primitives. These relations include collinearity, cocolority, coplanarity and symmetry. Of these relations, collinearity, cocolority and symmetry are defined for 2D as well as 3D primitives whereas by definition, coplanarity is meaningful only for 3D primitives. Table 1 summarizes the relations and on which dimension they are defined.

Relation	2D	3D
co-planarity	×	✓
co-colority	✓	✓
collinearity	✓	✓
symmetry	✓	✓

Table 1: The relations and in which dimension they are defined.

This paper does not focus on any specific application domain but provides a technically detailed definition of these relations that are usually not described in such detail in publications making use of them.

The paper is organized as follows: In section 2, we briefly introduce our visual features, namely primitives. In section 3, we describe our definitions of perceptual relations between the visual primitives. In section 5, we conclude the paper.

## 2 Primitives

Numerous feature detectors exist in the literature (see [30] for a review). Each feature based approach can be divided into an interest point detector (e.g. [14, 4]) and a descriptor describing a local patch of the image at this location, that can be based on histograms (e.g. [6, 30]), spatial frequency [20], local derivatives [15, 11, 1] steerable filters [12], or invariant moments ([28]). In [30] these different descriptors have been compared, showing a best performance for SIFT-like descriptors.

The primitives we will be using in this work are local, multi-modal edge descriptors that were introduced in [25]. In contrast to the above mentioned features these primitives focus on giving a semantically and geometrically meaningful description of the local image patch. The importance of such a semantic grounding of features for a general purpose vision front-end, and the relevance of edge-like structures for this purposes were discussed in [9].

The primitives are extracted sparsely at locations in the image that are the most likely to contain edges. This likelihood is computed using the intrinsic dimensionality measure proposed in [22]. The sparseness is assured using a classical winner take all operation, insuring that the generative patches of the primitives do not overlap (for details, see [26]). Each of the primitive encodes the image information contained by a local image patch. Multi-modal information is gathered from this image patch, including the position  $\mathbf{m}$  of the centre of the patch, the orientation  $\theta$  of the edge, the phase  $\omega$  of the signal at this point, the colour  $\mathbf{c}$  sampled over the image patch on both sides of the edge and the local optical flow  $\mathbf{f}$ . Consequently a local image patch is described by the following multi-modal vector:

$$\boldsymbol{\pi} = (\mathbf{m}, \theta, \omega, \mathbf{c}, \mathbf{f}, \rho)^T, \quad (1)$$

that we will name *2D primitive* in the following.

Note that these primitives are of lower dimensionality than, e.g., SIFT (10 vs. 128) and therefore suffer of a lesser distinctiveness. Nonetheless, as shown in [34] that they are distinctive enough for a reliable stereo matching if the epipolar geometry of the cameras is known. Furthermore, their semantic in terms of geometric and appearance based information allow for a good description of the scene content. It has been previously argued in [9] that edge pixels contain all important information in an image. As a consequence, the ensemble of all primitives extracted from an image describe the shapes present in this image.

Advantageously, the rich information carried by the 2D-primitives can be reconstructed in 3D, providing a more complete scene representation. Having geometrical meaning for the primitive allows to describe the relation between proximate primitives in terms of perceptual grouping.

In a stereo scenario 3D primitives can be computed from the correspondences of 2D primitives (see figure 1 and [34]):

$$\boldsymbol{\Pi} = (\mathbf{M}, \Theta, \boldsymbol{\Omega}, \mathbf{C})^T, \quad (2)$$

such that we have a projection relation:

$$\mathcal{P} : \boldsymbol{\Pi} \rightarrow \boldsymbol{\pi}. \quad (3)$$

## 3 Relations

In this section, we present collinearity, cocolority, coplanarity and symmetry relations that are defined on our visual features.

### 3.1 Collinearity in 2D and 3D

As the primitives are local contour descriptors, scene contours are expected to be represented by strings of primitives that are locally close to collinear. In the following, we will explain methods for grouping 2D and 3D primitives into contours.



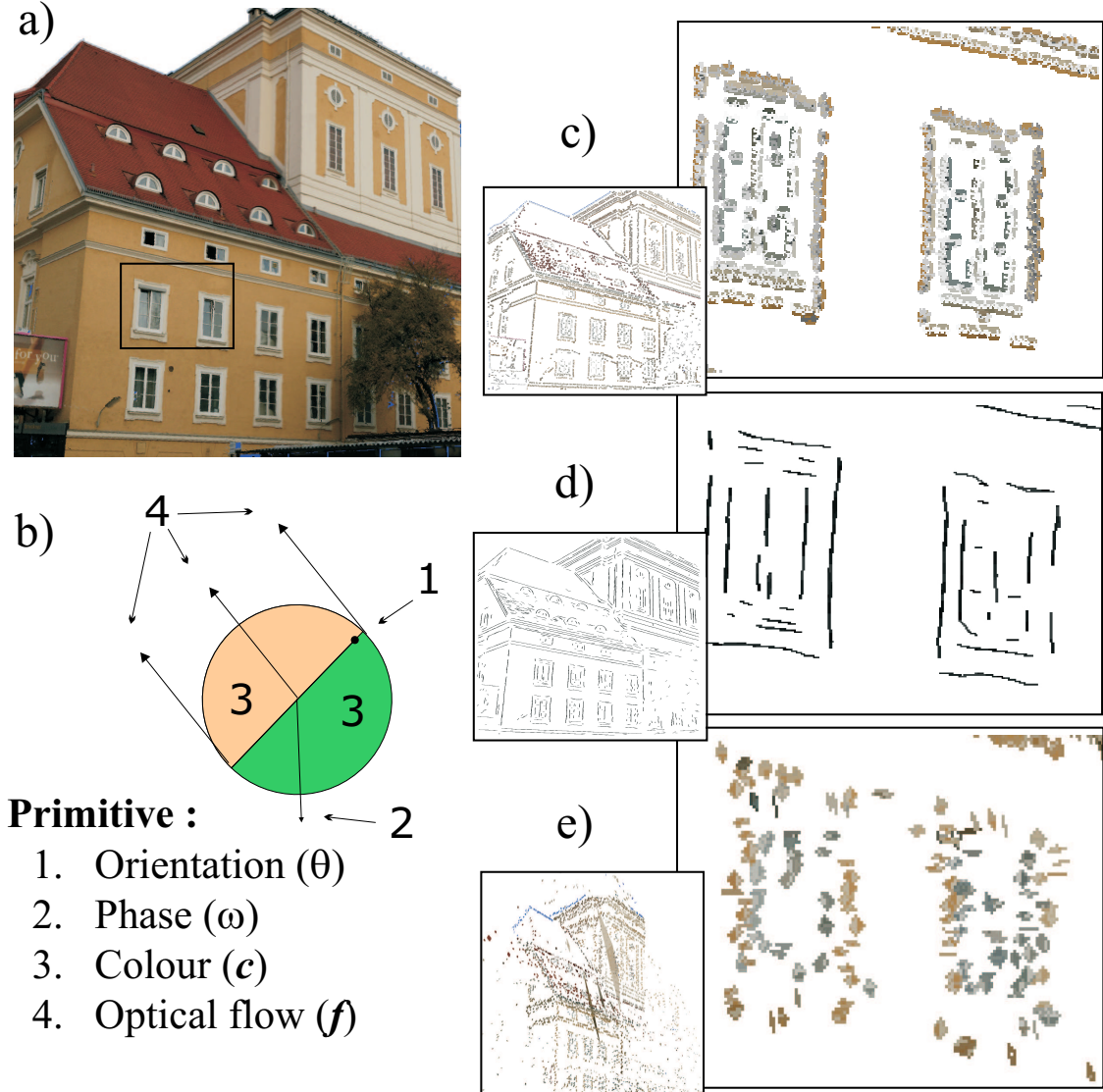


Figure 1: Illustration of the primitive extraction process from a video sequence. The 2D-primitives extracted from the input image (a) (see section 2), and finally the 3D-primitives reconstructed from the stereo-matches as described as described in [34]. **(a)** An example input image. **(b)** A graphic description of the 2D-primitives. **(c)** A magnification of the image representation. **(d)** Perceptual grouping of the primitives as described in [34]. **(e)** The reconstructed 3D entities. Note that the structure reconstructed is quite far from the cameras, leading to a certain imprecision in the reconstruction of the 3D-primitives. A simple scheme addressing this problem is described in [34].

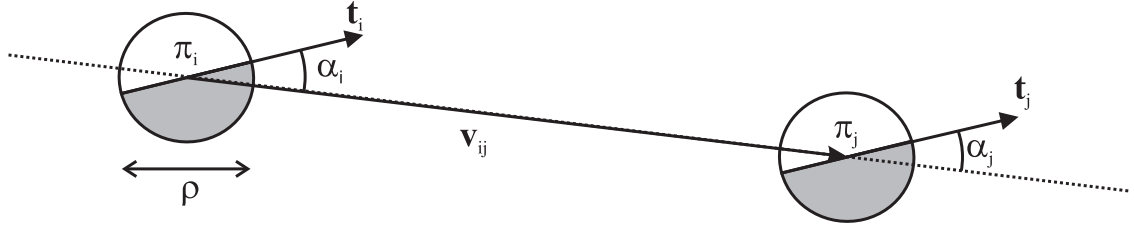


Figure 2: Illustration of the values used for the collinearity computation. If we consider two primitives  $\pi_i$  and  $\pi_j$ , then the vector between the centres of these two primitives is written  $\mathbf{v}_{ij}$ , and the orientations of the two primitives are designated by the vectors  $\mathbf{t}_i$  and  $\mathbf{t}_j$ , respectively. The angle formed by  $\mathbf{v}_{ij}$  and  $\mathbf{t}_i$  is written  $\alpha_i$ , and between  $\mathbf{v}_{ij}$  and  $\mathbf{t}_j$  is written  $\alpha_j$ .  $\rho$  is the radius of the image patch used to generate the primitive.

### 3.1.1 Collinearity in 2D

In the following,  $c(l_{i,j})$  refers to the likelihood for two primitives  $\pi_i$  and  $\pi_j$  to be *linked*: i.e. grouped to describe the same contour.

Position and orientation of primitives are intrinsically related. As primitives represent local edge estimators, their positions are points along the edge, and their orientation can be seen as a tangent at such a point. The estimated likelihood of the contour described by those tangents is based upon the assumption that simpler curves are more likely to describe the scene structures, and highly jagged contours are more likely to be manifestations of erroneous and noisy data.

Therefore, for a pair of primitives  $\pi_i$  and  $\pi_j$  in image  $\mathcal{I}$ , we can formulate the likelihood for these primitives to describe the same contour as a combination of three basic constraints on their relative position and orientation — see [34].

**Proximity** ( $c_p[l_{i,j}]$ ): A contour is more likely if it is described by a dense population of primitives. Large holes in the primitive description of the contour is an indication that there are two contours which are collinear yet different. The proximity constraint is defined by the following equation:

$$c_p[l_{i,j}] = 1 - e^{-\max\left(1 - \frac{\|\mathbf{v}_{i,j}\|}{\rho\tau}, 0\right)}, \quad (4)$$

where  $\rho$  stands for the size of the receptive field of the primitives in pixels;  $\rho\tau$  is the size of the neighbourhood considered in pixels; and,  $\|\mathbf{v}_{i,j}\|$  is the distance in pixels separating the centres of the two primitives.

**Collinearity** ( $c_{co}[l_{i,j}]$ ): A contour is more likely to be linear, or to form a shallow curve rather than a sharp one. A sharp curve might be an indication of two intersecting or occluding contours.

$$c_{co}[l_{i,j}] = 1 - \left| \sin \left( \frac{|\alpha_i| + |\alpha_j|}{2} \right) \right|, \quad (5)$$

where  $\alpha_i$  and  $\alpha_j$  are the angles between the line joining the two primitives centres and the orientation of, respectively,  $\pi_i$  and  $\pi_j$ .

**Co-circularity** ( $c_{ci}[l_{i,j}]$ ): A contour is more likely to have a continuous, or smoothly changing curvature, rather than a varying one. An unstable curvature is an indicator of a noisy, erroneous or under-sampled contour, all of which are unreliable.

$$c_{ci}[l_{i,j}] = 1 - \left| \sin \left( \frac{\alpha_i + \alpha_j}{2} \right) \right|, \quad (6)$$

**Geometric Constraint ( $\mathbf{G}_{i,j}$ ):** The combination of those three criteria provided above forms the following *geometric* affinity measure:

$$\mathbf{G}_{i,j} = \sqrt[3]{c_e[l_{i,j}] \cdot c_{co}[l_{i,j}] \cdot c_{ci}[l_{i,j}]}, \quad (7)$$

where  $\mathbf{G}_{i,j}$  is the geometric affinity between two primitives  $\pi_i$  and  $\pi_j$ . This affinity represents the likelihood that two primitives  $\pi_i$  and  $\pi_j$  are part of an actual contour of the scene.

**Multi-modal Constraint ( $\mathbf{M}_{i,j}$ ):** The geometric constraint offers a suitable estimation of the likelihood of the curve described by the pair of primitives. Other modalities of the primitives allow inferring more about the qualities of the physical contour they represent. The colour, phase and optical flow of the primitives further define the properties of the contour, and thus consistency constraints can also be enforced over those modalities. Effectively, the less difference there is between the modalities of two primitives, the more likely that they are expressions of the same contour. In [7], it is already proposed that the intensity can be used as a cue for perceptual grouping; our definition goes beyond this proposal by using a combination of the phase, colour and optical flow modalities of the primitives to decide if they describe the same contour:

$$\mathbf{M}_{i,j} = w_\omega c_\omega[l_{i,j}] + w_c c_c[l_{i,j}] + w_f c_f[l_{i,j}], \quad (8)$$

where  $c_\omega$  is the phase criterion,  $c_c$  the colour criterion and  $c_f$  the optical flow criterion. Each of the three  $w_\omega$ ,  $w_c$  and  $w_f$  is the relative scaling for each modality, with  $w_\omega + w_c + w_f = 1$ .

**Primitive Affinity ( $\mathbf{A}_{i,j}$ ):** The overall affinity between all primitives in an image is formalised as a matrix  $\mathbf{A}$ , where  $\mathbf{A}_{i,j}$  holds the affinity between the primitives  $\pi_i$  and  $\pi_j$ . We define this affinity from equations 7 and 8, such that (1) two primitives complying poorly with the good continuation rule have an affinity close to zero; and (2) two primitives complying with the good continuation rule yet strongly dissimilar will have only an average affinity. The affinity is formalised as follows:

$$c(l_{i,j}) = \mathbf{A}_{i,j} = \sqrt{\mathbf{G} (\alpha \mathbf{G}_{i,j} + (1 - \alpha) \mathbf{M}_{i,j})}, \quad (9)$$

where  $\alpha$  is the weighting of geometric and multi-modal (*i.e.* phase, colour and optical flow) information in the affinity. A setting of  $\alpha = 1$  implies that only geometric information (proximity, collinearity and co-circularity) is used, while  $\alpha = 0$  means that geometric and multi-modal information are evenly mixed.

### 3.1.2 Collinearity in 3D

Collinearity in 3D is more difficult to define. Due to the inaccuracy in stereo-reconstruction of 3D position and orientation, it is impossible to apply strong alignment constraints such as the ones we applied in the 2D case. Consequently we will define 3D collinearity as follows:

**Definition 1** *Two 3D-primitives  $\Pi_i$  and  $\Pi_j$  are said collinear if the 2D-primitives  $\pi_i^x$  and  $\pi_j^x$  they project onto the camera plane  $x$  (defined by a projection relation  $\mathcal{P}^x : \Pi_k \rightarrow \pi_k$ ) are all collinear (according to the definition of 2D-primitive collinearity presented above).*

and therefore in the standard case where we have two stereo cameras labelled  $l$  and  $r$  we have the following relation:

$$c(L_{i,j}) = c(l_{i,j}^l) \cdot c(l_{i,j}^r). \quad (10)$$

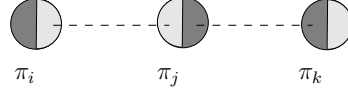


Figure 3: Co-colority of three 2D primitives  $\pi_i$ ,  $\pi_j$  and  $\pi_k$ . In this case,  $\pi_i$  and  $\pi_j$  are cocolor, so are  $\pi_i$  and  $\pi_k$ ; however,  $\pi_j$  and  $\pi_k$  are not cocolor.

### 3.2 Cocolority in 2D and 3D

Two spatial primitives  $\Pi_i$  and  $\Pi_j$  are co-color iff their parts that face each other have the same color. In the same way as collinearity, co-colority of two spatial primitives  $\Pi_i$  and  $\Pi_j$  is computed using their 2D projections  $\mathcal{P}\Pi_i = \pi_i$  and  $\pi_j$ . We define the co-colority of two 2D primitives  $\pi_i$  and  $\pi_j$  as:

$$\text{coc}(\pi_i, \pi_j) = 1 - \mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j),$$

where  $\mathbf{c}_i$  and  $\mathbf{c}_j$  are the RGB representation of the colors of the parts of the primitives  $\pi_i$  and  $\pi_j$  that face each other; and,  $\mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j)$  is Euclidean distance between RGB values of the colors  $\mathbf{c}_i$  and  $\mathbf{c}_j$ . In Fig. 3, a pair of co-color and not co-color primitives are shown.

Euclidean color distance  $d_c$  is a simple one compared to color distance metrics developed by different institutes like International Commission on Illumination (CIE). Such metrics are developed to match our perception of colour and are computationally expensive (see, *e.g.*, [38]). For our purposes, Euclidean distance between RGB values is sufficient and can be replaced by a more complicated distance metric, if desired.

3D co-colority is defined as follows:

**Definition 2** *Two 3D-primitives  $\Pi_i$  and  $\Pi_j$  are said cocolor if the 2D-primitives  $\pi_i^x$  and  $\pi_j^x$  they project onto the camera plane  $x$  (defined by a projection relation  $\mathcal{P}^x : \Pi_k \rightarrow \pi_k$ ) are co-color (according to the definition of 2D-primitive cocolority presented above).*

### 3.3 Coplanarity

According to [37],

a set of points in space is coplanar if the points all lie in a geometric plane. For example, three points are always coplanar; but four points in space are usually not coplanar.

Although the definitions are more or less the same, there are different ways to *check* the coplanarity of a set of points [36, 37]. For a set of  $n$  points  $\mathbf{x}_1 \dots \mathbf{x}_n$  where  $\mathbf{x}_i = (x_i, y_i, z_i)$ , the following methods can be adopted:

- For  $n = 4$ ,  $\mathbf{x}_1 \dots \mathbf{x}_n$  are coplanar
  - iff the volume of the tetrahedron defined by them is 0 [36], *i.e.*,

$$\begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} = 0. \quad (11)$$

- iff the pair of lines determined by the four points are not skew [36]:

$$(\mathbf{x}_3 - \mathbf{x}_1) \cdot [(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_4 - \mathbf{x}_3)] = 0. \quad (12)$$

- iff  $\mathbf{x}_4$  is on the plane defined by  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ :

$$d(\mathbf{x}_4, P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)) = 0, \quad (13)$$

where  $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$  is the plane defined by  $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ , and  $d(\mathbf{x}, \mathbf{p})$  is the distance between point  $\mathbf{x}$  and plane  $\mathbf{p}$ .

- For  $n > 4$ ,  $\mathbf{x}_1 \dots \mathbf{x}_n$  are coplanar iff point-plane distances of  $\mathbf{x}_4 \dots \mathbf{x}_n$  to the plane defined by  $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$  are all zero:

$$\sum_{i=4}^n d(\mathbf{x}_i, P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)) = 0. \quad (14)$$

### 3.3.1 Coplanarity of bounded planes

A bounded plane  $\mathbf{p}^b$  is part of the plane  $\mathbf{p}$  with a certain size  $\mathbf{s}$  and position  $\mathbf{x}$ . In other words,  $\mathbf{p}^b$  is equivalent to  $(\mathbf{n}, \mathbf{x}, \mathbf{s})$  where  $\mathbf{n}, \mathbf{x}, \mathbf{s}$  are respectively the normal (*i.e.*, orientation), position (*i.e.*, center) and the size of the bounded plane.

As suggested in [17], two bounded planes  $\mathbf{p}_1^b, \mathbf{p}_2^b$  are coplanar if:

$$(\alpha(\mathbf{n}_1, \mathbf{n}_2) < T_\alpha) \wedge \left( \frac{d(\mathbf{x}_1, \mathbf{p}_2^b)}{d(\mathbf{x}_1, \mathbf{x}_2)} < T_d \right), \quad (15)$$

where  $\alpha(\mathbf{n}_1, \mathbf{n}_2)$  is the angle between the two orientations vectors  $\mathbf{n}_1$  and  $\mathbf{n}_2$ , and  $T_\alpha$  and  $T_d$  are the thresholds.

### 3.3.2 Coplanarity of 3D primitives

Two spatial primitives  $\Pi_i$  and  $\Pi_j$  are co-planar iff their orientation vectors lie on the same plane, *i.e.*:

$$\text{cop}(\Pi_i, \Pi_j) = 1 - |\text{proj}_{t_j \times v_{ij}}(t_i \times v_{ij})|, \quad (16)$$

where  $\mathbf{v}_{ij}$  is defined as the vector  $(\mathbf{M}_i - \mathbf{M}_j)$ ;  $t_i$  and  $t_j$  denote the vectors defined by the 3D orientations  $\Theta_i$  and  $\Theta_j$ , respectively; and  $\text{proj}_{\mathbf{u}}(\mathbf{a})$  is defined as:

$$\text{proj}_{\mathbf{u}}(\mathbf{a}) = \frac{\mathbf{a} \cdot \mathbf{u}}{\|\mathbf{u}\|^2} \mathbf{u}. \quad (17)$$

The co-planarity relation is illustrated in Fig. 4.

## 3.4 Symmetry in 2D and 3D

Two primitives are symmetric if they are located on two contours which are reflections of each other (see figure 5(a)). This reflective symmetry between two primitives can be measured by utilizing the angles between the orientations of the primitives and the line that joins the centers of the primitives.

Let  $v_{ij}$  denote the line joining the centers of the primitives,  $\pi_i$  and  $\pi_j$ , and also  $\phi_{ij}$  and  $\phi_{ji}$  be the angles between  $v_{ij}$  and the lines defined by the orientations of  $\pi_i$  and  $\pi_j$ , respectively (see figure 5). Then, two 2D primitives  $\pi_i$  and  $\pi_j$  can be considered symmetric, if  $\phi_{ij} = \phi_{ji}$  with a symmetry axis  $a_{ij}$  defined as follows:

$$a_{ij} = \begin{cases} L(c_{ij}; \theta_i) & \text{if } \theta_i = \theta_j, \\ L(c_{ij}; \alpha_{ij}), & \text{otherwise,} \end{cases} \quad (18)$$

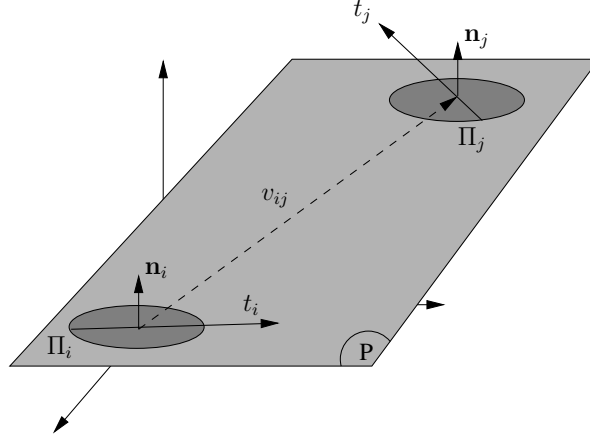


Figure 4: Co-planarity of two 3D primitives  $\Pi_i$  and  $\Pi_j$ .  $t_i$  and  $t_j$  denote the vectors defined by the 3D orientations  $\Theta_i$  and  $\Theta_j$ , respectively.

where  $L(x; \theta)$  is a line that goes through a point  $x$  with orientation  $\theta$ ;  $\text{int}(l_k, l_m)$  is the intersection point of two lines denoted by  $l_k$  and  $l_m$ ;  $c_{ij}$  is defined as the mid-point of  $v_{ij}$  (i.e.,  $(\mathbf{m}_i + \mathbf{m}_j)/2$ ); and,  $\alpha_{ij}$  is the angle of the line that joins the points  $c_{ij}$  and  $\text{int}(L(\mathbf{m}_i; \theta_i), L(\mathbf{m}_j; \theta_j))$ .

The symmetry axis  $a_{ij}$  is undefined if the primitive orientations  $\theta_i$  and  $\theta_j$ , and  $v_{ij}$  are all parallel, which is the case when both primitives are located on the same linear segment of a contour. This is the case for  $\pi_j$  and  $\pi_k$  in figure 5(b) and 5(c). If the symmetry axis  $a_{ij}$  is undefined, a primitive pair should not be regarded as symmetric, but collinear.

Figure 5 illustrates a few symmetric and non-symmetric primitives. In figure 5(b) and 5(c), as the primitives  $\pi_j$  and  $\pi_k$  are on the same contour,  $a_{ij}$  is parallel with the primitive orientations  $\theta_j$ ,  $\theta_k$  and  $v_{jk}$ .

Taking collinearity into account, symmetry between two primitives  $\pi_i$  and  $\pi_j$  is defined as follows:

$$\text{sym}(\pi_i, \pi_j) = \begin{cases} 0 & \text{if } c_{co}[l_{i,j}] > T_c, \\ 1 - |\sin(\phi_{ij} - \phi_{ji})| & \text{otherwise,} \end{cases} \quad (19)$$

where  $c_{co}[l_{i,j}]$  is the collinearity relation and  $T_c$  is a threshold, determining if  $\pi_i$  and  $\pi_j$  are collinear.

Like collinearity and co-colority, the symmetry of two 3D primitives  $\Pi_i$  and  $\Pi_j$  is computed using their 2D projections  $\pi_i$  and  $\pi_j$ :

**Definition 3** Two 3D-primitives  $\Pi_i$  and  $\Pi_j$  are said to be symmetric if the 2D-primitives  $\pi_i^x$  and  $\pi_j^x$  they project onto the camera plane  $x$  (defined by a projection relation  $\mathcal{P}^x : \Pi_k \rightarrow \pi_k$ ) are symmetric (according to the definition of 2D-primitive symmetry presented above).

## 4 Results

In figure 6, the coplanarity, cocolority and collinearity relations are shown for two different example scenes shown in figure 6(a) and (b). The results are from our 3D display tool called *Wanderer*, and for computational reasons, 3D primitives are shown in squares. The relations are displayed only for a primitive which is selected with the mouse as showing relations between all primitives disables visibility.

From the figure we see that coplanarity is a more common relation than cocolority or collinearity. This suggests that coplanarity alone is not directly usable for analysis or applications in 3D, and it needs to be accompanied with other relations as proposed and utilized in [2, 16].

## 5 Conclusion

In this paper, we presented cocolority, coplanarity, collinearity and symmetry relations defined on multi-modal visual features, called primitives.

Such relations have been utilized in different perceptual organization problems as well as analysis of how the natural scenes are structured (see, *e.g.*, ([3, 8, 10, 13, 16, 17, 21, 27, 31, 34, 39])), and the importance of such relations, as well as their psychophysical and biological plausibility have been acknowledged in the literature (see, *e.g.*, [18, 19, 35]).

## 6 Acknowledgments

We would like to thank Florentin Wörgötter and Daniel Aarno for their fruitful contributions. This work is supported by the Drivscio and the PACO+ projects.

## References

- [1] A. Baumberg. Reliable Feature Matching across Widely Separated Views. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 774–781, 2000.
- [2] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Model-independent grasping initializing object-model learning in a cognitive architecture. *IEEE International Conference on Robotics and Automation (ICRA), Workshop: From features to actions - Unifying perspectives in computational and robot vision*, 2007.
- [3] E. Brunswik and J. Kamiya. Ecological cue-validity of ‘proximity’ and of other Gestalt factors. *American Journal of Psychology*, LXVI:20–32, 1953.
- [4] Cordelia Schmid and Roger Mohr and Christian Baukhage. Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.
- [5] J. S. D. Aarno, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Early reactive grasping with second order 3d feature relations. *IEEE Conference on Robotics and Automation (submitted)*, 2007.
- [6] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.
- [7] J. Elder and R. Goldberg. Inferential reliability of contour grouping cues in natural images. *Perception Supplement*, 27, 1998.
- [8] J. Elder and R. Goldberg. Ecological statistics of Gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2(4):324–353, 8 2002.
- [9] J. H. Elder. Are edges incomplete ? *International Journal of Computer Vision*, 34:97–122, 1999.
- [10] J. H. Elder, A. Krupnik, and L. A. Johnston. Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(25):1–14, 2003.
- [11] Frederik Schaffalitzky and Andrew Zisserman. Multi-view Matching for Unordered Image Sets, or “How Do I Organize My Holiday Snaps?”. *Lecture Notes in Computer Science*, 2350:414–431, 2002. in Proceedings of the BMVC02.
- [12] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE-PAMI*, 13(9):891–906, 1991.

- [13] W. E. L. Grimson. A Computational Theory of Visual Surface Interpolation. *Royal Society of London Philosophical Transactions Series B*, 298:395–427, Sept. 1982.
- [14] C. G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.
- [15] J. J. Koenderink and A. J. van Doorn. Representation of Local Geometry in the Visual System. *Biological Cybernetics*, 55:367–375, 1987.
- [16] S. Kalkan, F. Wörgötter, and N. Krüger. Depth prediction at homogeneous image structures. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-2, 2007.
- [17] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of second-order relations of 3d structures. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.
- [18] K. Koffka. *Principles of Gestalt Psychology*. Lund Humphries, London, 1935.
- [19] K. Köhler. *Gestalt Psychology: An introduction to new concepts in psychology*. New York: Liveright, 1947.
- [20] P. Kovési. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1(3):1–26, 1999.
- [21] N. Krüger. Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129, 1998.
- [22] N. Krüger and M. Felsberg. A continuous formulation of intrinsic dimension. *Proceedings of the British Machine Vision Conference*, pages 261–270, 2003.
- [23] N. Krüger, M. V. Hulle, and F. Wörgötter. Ecovision: Challenges in early-cognitive vision. *International Journal of Computer Vision*, accepted.
- [24] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5):417–428, 2004.
- [25] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *Interdisciplinary Journal of Artificial Intelligence & the Simulation of Behaviour, AISB Journal*, 1(5):417–427, 2004.
- [26] N. Krüger, N. Pugeault, and F. Wörgötter. Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-4, 2007.
- [27] N. Krüger and F. Wörgötter. Statistical and deterministic regularities: Utilisation of motion and grouping in biological and artificial visual systems. *Advances in Imaging and Electron Physics*, 131:82–147, 2004.
- [28] Luc Van Gool and Theo Moons and Dorin Ungureanu. Affine / Photometric Invariants for Planar Intensity Patterns. *Lecture Notes In Computer Science*, 1064:642–651, 1996. in Proceedings of the 4th European Conference on Computer Vision — Volume 1.
- [29] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. Freeman, 1977.
- [30] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
- [31] N. Pugeault, N. Krüger, and F. Wörgötter. A non-local stereo similarity based on collinear groups. *Proceedings of the Fourth International ICSC Symposium on Engineering of Intelligent Systems*, 2004.



- [32] N. Pugeault, N. Krüger, and F. Wörgötter. Rigid body motion estimation in an early cognitive vision framework. In *IEEE Advances In Cybernetic Systems*, 2006.
- [33] N. Pugeault, F. Wörgötter, , and N. Krüger. Disambiguation .....
- [34] N. Pugeault, F. Wörgötter, and N. Krüger. Multi-modal scene reconstruction using perceptual grouping constraints. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06)*, 2006.
- [35] S. Sarkar and K. Boyer. *Computing Perceptual Organization in Computer Vision*. World Scientific, 1994.
- [36] E. W. Weisstein. Coplanar. from mathworld—a wolfram web resource, 2006. <http://mathworld.wolfram.com/Coplanar.html>.
- [37] Wikipedia. Coplanarity — wikipedia, the free encyclopedia, 2006. <http://en.wikipedia.org/w/index.php?title=Coplanarity&oldid=37490165>.
- [38] X. Zhang and B. A. Wandell. Color image fidelity metrics evaluated using image distortion maps. *Signal Processing*, 70(3):201–214, 1998.
- [39] S. C. Zhu. Embedding gestalt laws in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1170–1187, 1999.

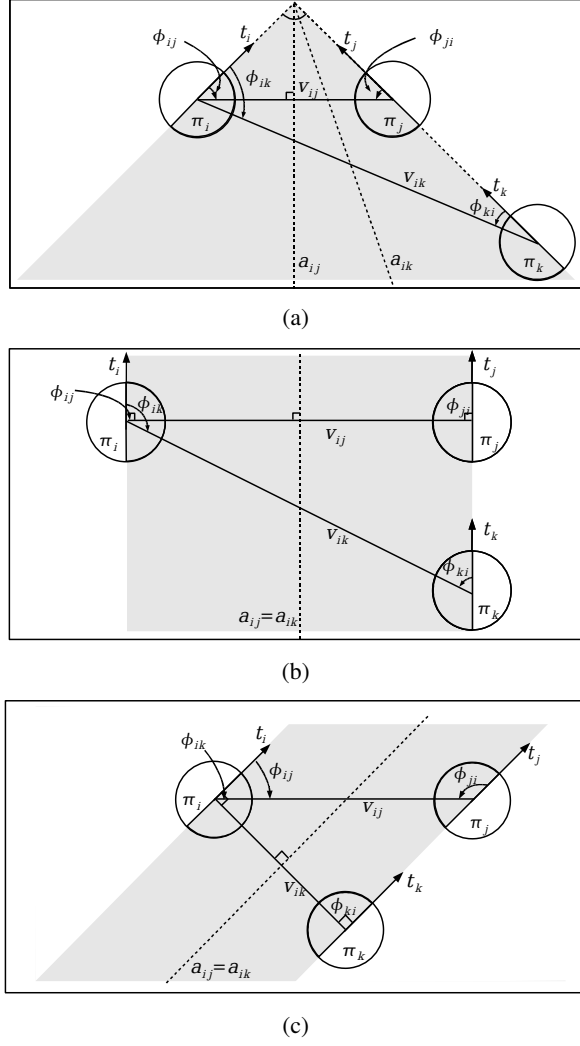


Figure 5: Illustration of the definition of symmetry.  $t_i$ ,  $t_j$  and  $t_k$  denote the vectors defined by the orientations  $\theta_i$ ,  $\theta_j$  and  $\theta_k$ , respectively. Primitives  $\pi_i$  and  $\pi_j$  are symmetric in (a) and (b), but not in (c).  $\pi_i$  and  $\pi_k$  are symmetric in (c), but not in (a) or (b).

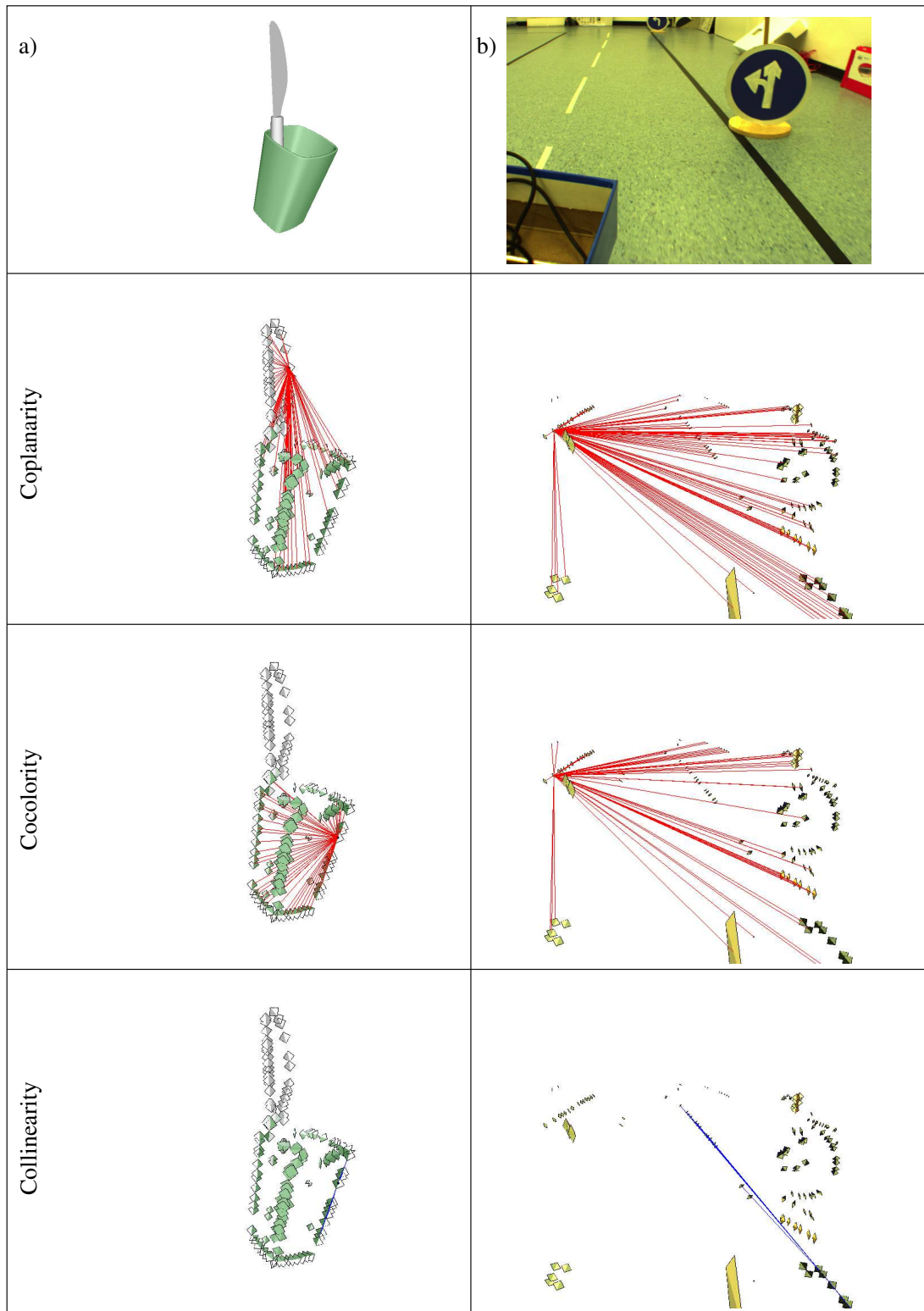


Figure 6: The coplanarity, cocolority and collinearity relations on two different examples shown in (a) and (b). The results are from our 3D display tool called *Wanderer*, and for the sake of speed, 3D primitives are shown in squares. The relations are shown only for a selected primitive as showing relations between all primitives disables visibility.

**Appendix F**  
***Depth Prediction at Homogeneous Image structures.***  
S. Kalkan, F. Wörgötter and N. Krüger  
( *Technical Report 2007 – 2* )

Robotics Group  
The Maersk Mc-Kinney Moller Institute  
University of Southern Denmark

---

Technical Report no. 2007 – 2

---

**Depth Prediction at Homogeneous Image  
structures**

Sinan Kalkan, Florentin Wörgötter, Norbert Krüger

February 10, 2007

Title	Depth Prediction at Homogeneous Image structures
	Copyright © 2007 Sinan Kalkan, Florentin Wörgötter, Norbert Krüger. All rights reserved.
Author(s)	Sinan Kalkan, Florentin Wörgötter, Norbert Krüger
Publication History	

## Abstract

Depth at homogeneous or weakly-textured image areas is difficult to obtain because such image areas suffer the well-known correspondence problem. In this paper, we propose a voting model that predicts the depth at such image areas from the depth of bounding edge-like structures. The depth at edge-like structures is computed using a feature-based stereo algorithm, and is used to vote for the depth of homogeneous image areas. We show the results of our ongoing work on different scenarios.

## 1 Introduction

Extraction of 3D structure from 2D images is realized by utilizing a set of inverse problems that include structure from motion, stereo vision, shape from shading, linear perspective, texture gradients and occlusion [3]. These cues can be classified as pictorial, or monocular, (such as shading, utilization of texture gradients or linear perspective) and multi-view (like stereo and structure from motion). Depth cues which make use of multiple views require correspondences between different 2D views of the scene. In contrast, pictorial cues use statistical and geometrical relations in one image to make statements about the underlying 3D structure. Many surfaces have only weak texture or no texture at all, and as a consequence, the *correspondence problem is very hard or not at all resolvable for these surfaces*. Nevertheless, humans are able to reconstruct 3D information for these surfaces, too. Existing psychophysical experiments (see, e.g., [2, 4]) and computational theories (see, e.g., [1, 6, 26]) suggest that in the human visual system, *an interpolation process* is realized that starting with the local analysis of edges, corners and textures, computes depth also in areas where correspondences cannot easily be found.

In this paper, we are interested in prediction of depth at homogeneous image patches (called *monos* in this paper) from the depth of the edges in the scene using a voting model. We start by creating a representation of the input stereo images in terms of local image patches corresponding to edge-like structures and monos (as introduced in [15] and section 2, and described in detail in [16]). The depth at edge-like patches is extracted using feature-based stereo computation between the two images (using the method introduced in [22]). The depth that is extracted at the bounding edge-like patches of a mono using stereo votes for its depth.

We would like to distinguish *depth prediction* from *surface interpolation* because surface interpolation assumes that there is already a dense depth map of the scene available in order to be able to estimate the 3D orientation at points (see, e.g., [6, 7, 8, 18, 19, 25, 26]) whereas our understanding of depth prediction makes use of only 3D line-orientations at edge-segments which are computed using a feature-based stereo proposed in [22].

A typical scenario that our model is designed for is shown in figure 1 where an input stereo pair and the stereo data (computed using [22]) are displayed. We see that computed stereo information has strong outliers which prohibit a *surface interpolation* method as it is not possible to differentiate between the outliers and the reliable stereo information. Moreover, the stereo information that should be reliable at the edges of the road turn out not to share a common surface nor the same 3D line (see figure 1(c)). Applying a surface interpolation method on such input data is expected to lead to a wrong road surface prediction. In this paper, we will show that our depth prediction method is able to cope with such strong outliers.

### 1.1 Related studies

It is fair to count the early works of Grimson [6] as the pioneers of surface interpolation. In [6], Grimson proposed fitting square Laplacian functionals to surface orientations at existing 3D points utilizing a *surface consistency constraint* called 'no news is good news'. The constraint argues that if two image points do not have a contrast difference in-between, then they can be assumed to be on the same 3D surface (see [11] for

a quantification of this assumption). This work is extended in [7] with use of shading information. [6, 7] assume that surface information is available, and the input 3D points are dense enough for second order differentiation.

In [1], surface orientation at homogeneous image areas is recovered by *interpreting line drawings*. Lines are classified as extremal or discontinuity by making use of the junction labels and global relations like symmetry and parallism. They assume that (1) extremal points (the boundaries of the objects) in an image correspond to surface orientations which are normal to the image curve and the line of sight, and that (2) discontinuities (lines other than extremal points) lead to surface orientations which are normal to space curve. The underlying assumptions of [1] are that (1) a clean contour of the scene is provided, and that (2) the object is separated from the background. Moreover, the results provided in [11] suggest that it may not be a good idea to assume that edges correspond to only certain types of surface orientations. [21, 24, 27, 28] are similar to [1] as far as our paper is concerned.

In [8], 3D points with surface orientation are interpolated using a perceptual constraint called *co-surfacity* which produces a 3D association field (which is called Diabolo field by the authors) similar to the association field used in 2D perceptual contour grouping studies. If the points do not have 3D orientation, they estimate the 3D orientation first and then apply the surface interpolation step. In [18, 19], it is argued that stereo matching and surface interpolation should not be sequential but rather simultaneous. For this, they employ the following steps: (1) Normalized-cross correlation and edge-based stereo are computed. (2) The disparities are combined and disparities corresponding to inliers, surfaces and surface discontinuities are marked using tensor voting. (3) Surfaces are extracted using marching cubes approach. At this stage, surfaces are over the boundaries. (4) At the last step, over-boundary surfaces are trimmed. They assume sphere as their surface model when interpolating surface orientations.

Our method is similar to shape from silhouette methods which try to estimate the 3D information from the occluding edges of a single object (see, *e.g.*, [13, 20]). As put forward in [20], these methods are limited to spherical objects, and the underlying principles are valid only for occluding edges.

In [25, 26], stereo is computed at different scales, and instead of collapsing the results of these different scales into a single layer of disparity estimation and then applying surface interpolation, surface interpolation is applied separately for each scale and the results are combined.

Our work is different from the above mentioned works in that:

- Our approach does not assume that the input stereo points are dense enough to compute their 3D orientation (this is why the authors of this paper prefer to distinguish between depth prediction and surface interpolation). Instead, our method relies on the 3D line-orientations of the edge segments

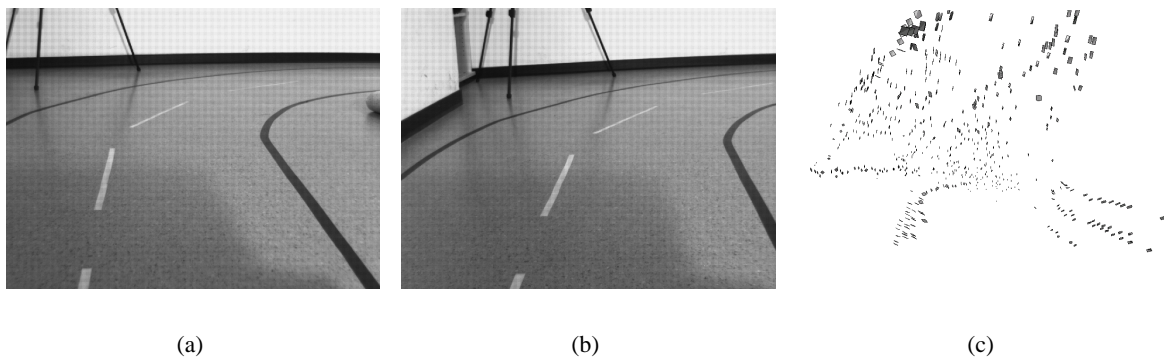


Figure 1: An input stereo pair ((a) and (b)) and how a feature-based stereo algorithm (taken from [22]) looks like (c).



which are extracted using a feature-based stereo algorithm (proposed in [22]).

- We employ a voting method like [18, 19] but is different, allowing long-range interactions in empty image areas, in order to predict *both* the depth and the surface orientation.

The paper is organized as follows: In section 2, we introduce how the images are represented in terms of local image patches. Section 3 describes the 2D and 3D relations between the local image patches that are utilized in the depth prediction process. Section 4 gives the outline of how the depth prediction is performed. In section 5, the results are presented and discussed. Finally, in section 6, the paper is concluded.

## 2 Visual Features

The visual features we utilize (called primitives in the rest of the paper) are local, multi-modal feature descriptors that were introduced in [15]. They are semantically and geometrically meaningful descriptions of local patches, motivated by the hyper-columnar structures in V1 ([9]).

An edge-like primitive can be formulated as:

$$\pi^e = (\mathbf{x}, \theta, \omega, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r), f), \quad (1)$$

where  $\mathbf{x}$  is the image position of the primitive;  $\theta$  is the 2D orientation;  $\omega$  represents the contrast transition;  $(\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$  is the representation of the color, corresponding to the left ( $\mathbf{c}_l$ ), the middle ( $\mathbf{c}_m$ ) and the right side ( $\mathbf{c}_r$ ) of the primitive; and,  $f$  is the optical flow extracted using Nagel-Enkelmann optic flow algorithm. As the underlying structure of an homogeneous image patch is different from that of an edge-like patch, a different representation is needed for homogeneous image structures (called *monos* in this paper):

$$\pi^m = (\mathbf{x}, \mathbf{c}), \quad (2)$$

where  $\mathbf{x}$  is the image position, and  $\mathbf{c}$  is the color of the mono.

See [17] for more information about these modalities and their extraction. Figure 2 shows extracted primitives for an example scene.

$\pi^e$  is a 2D feature which can be used to find correspondences in a stereo framework to create 3D primitives (as introduced in [14, 23]) with the following formulation:

$$\Pi^e = (\mathbf{X}, \Theta, \Omega, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)), \quad (3)$$

where  $\mathbf{X}$  is the 3D position;  $\Theta$  is the 3D orientation;  $\Omega$  is the phase (i.e., contrast transition); and,  $(\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$  is the representation of the color, corresponding to the left ( $\mathbf{c}_l$ ), the middle ( $\mathbf{c}_m$ ) and the right side ( $\mathbf{c}_r$ ) of the 3D primitive.

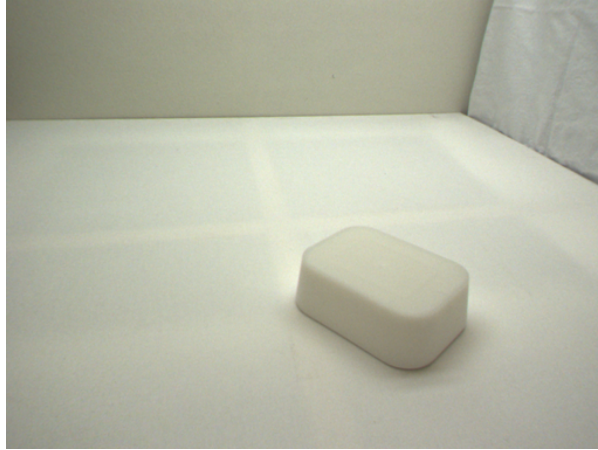
In this paper, we estimate the 3D representation  $\Pi^m$  of monos which stereo fails to compute:

$$\Pi^m = (\mathbf{X}, \mathbf{n}, \mathbf{c}), \quad (4)$$

where  $\mathbf{X}$  and  $\mathbf{c}$  are as in equation 2, and  $\mathbf{n}$  is the orientation (i.e., normal) of the plane that locally represents the mono.

## 3 Relations between Primitives

Sparse and symbolic nature of primitives allows the following relations to be defined on them. For more information about relations of primitives, see [10].



(a) Input image.



(b) Extracted primitives.

Figure 2: Extracted primitives (b) for the example image in (a).

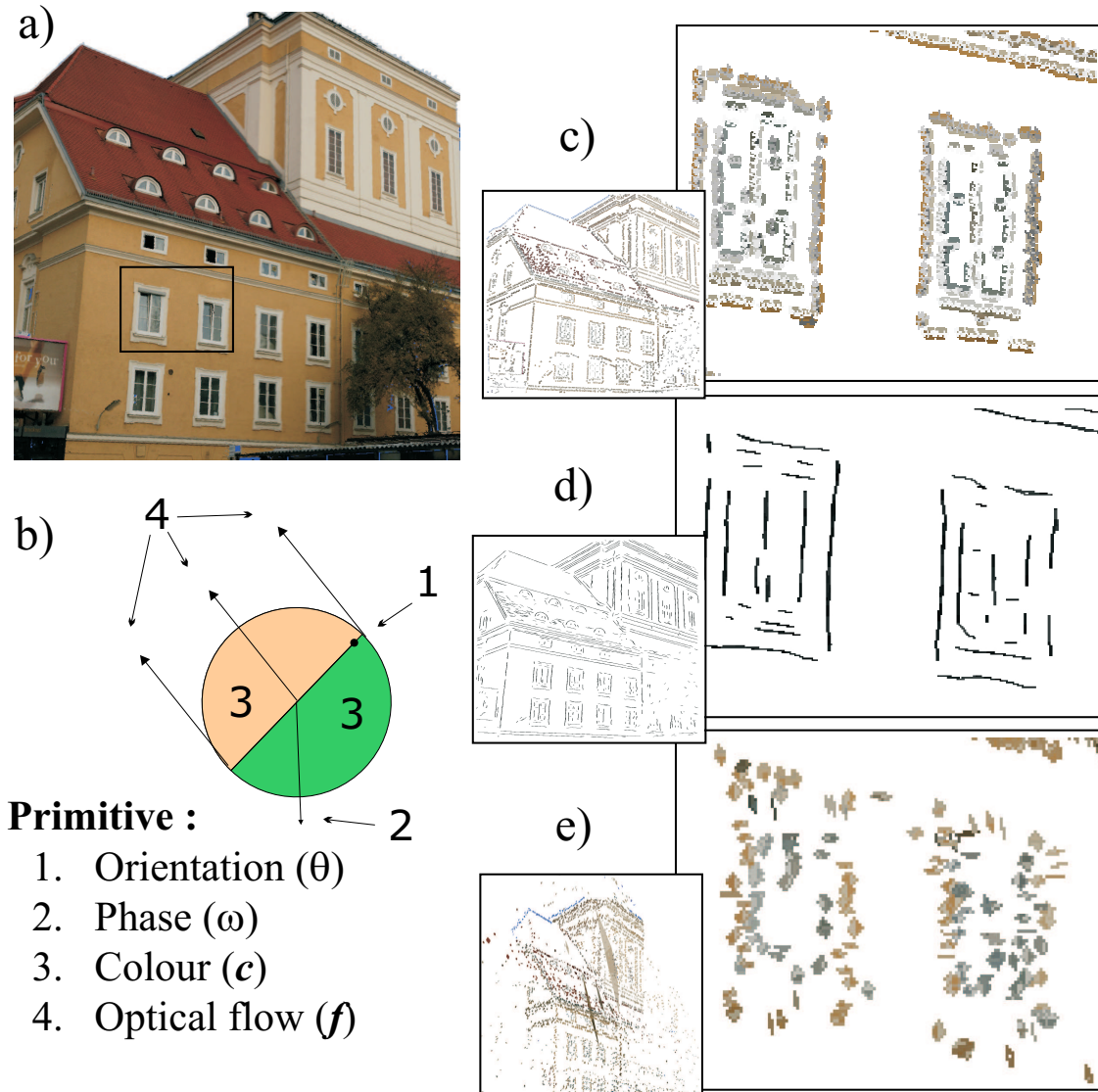


Figure 3: Illustration of the primitive extraction process from a video sequence. The 2D-primitives extracted from the input image (a) (see section 2), and finally the 3D-primitives reconstructed from the stereo-matches as described as described in [23]. **(a)** An example input image. **(b)** A graphic description of the 2D-primitives. **(c)** A magnification of the image representation. **(d)** Perceptual grouping of the primitives as described in [23]. **(e)** The reconstructed 3D entities. Note that the structure reconstructed is quite far from the cameras, leading to a certain imprecision in the reconstruction of the 3D-primitives. A simple scheme addressing this problem is described in [23].

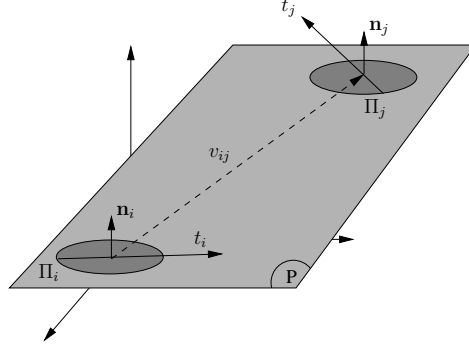


Figure 4: Co-planarity of two 3D primitives  $\Pi_i^e$  and  $\Pi_j^e$ .

### 3.1 Co-planarity

Two 3D edge primitives  $\Pi_i^e$  and  $\Pi_j^e$  are co-planar iff their orientation vectors lie on the same plane, i.e.:

$$\text{cop}(\Pi_i^e, \Pi_j^e) = 1 - |\mathbf{proj}_{t_j \times v_{ij}}(t_i \times v_{ij})|, \quad (5)$$

where  $v_{ij}$  is defined as the vector  $(\mathbf{X}_i - \mathbf{X}_j)$ ;  $t_i$  and  $t_j$  denote the vectors defined by the 3D orientations  $\Theta_i$  and  $\Theta_j$ , respectively; and,  $\mathbf{proj}_{\mathbf{u}}(\mathbf{a})$  is defined as:

$$\mathbf{proj}_{\mathbf{u}}(\mathbf{a}) = \frac{\mathbf{a} \cdot \mathbf{u}}{\|\mathbf{u}\|^2} \mathbf{u}. \quad (6)$$

The co-planarity relation is illustrated in Fig. 4.

### 3.2 Linear dependence

Two 3D primitives  $\Pi_i^e$  and  $\Pi_j^e$  are linearly dependent iff the *three* lines which are defined by (1) the 3D orientation of  $\Pi_i^e$ , (2) the 3D orientation of  $\Pi_j^e$  and (3)  $v_{ij}$  are identical. Due to uncertainty in the 3D reconstruction process, in this work, the linear dependence of two spatial primitives  $\Pi_i^e$  and  $\Pi_j^e$  is computed using their 2D projections  $\pi_i^e$  and  $\pi_j^e$ . We define the linear dependence of two 2D primitives  $\pi_i^e$  and  $\pi_j^e$  as:

$$\text{lin}(\pi_i^e, \pi_j^e) = |\mathbf{proj}_{v_{ij}} t_i| > Th \wedge |\mathbf{proj}_{v_{ij}} t_j| > Th, \quad (7)$$

where  $t_i$  and  $t_j$  are the vectors defined by the orientations  $\theta_i$  and  $\theta_j$ , respectively; and,  $Th$  is a threshold.

### 3.3 Co-colority

Two 3D primitives  $\Pi_i^e$  and  $\Pi_j^e$  are co-color iff their parts that face each other have the same color. In the same way as linear dependence, co-colority of two spatial primitives  $\Pi_i^e$  and  $\Pi_j^e$  is computed using their 2D projections  $\pi_i^e$  and  $\pi_j^e$ . We define the co-colority of two 2D primitives  $\pi_i^e$  and  $\pi_j^e$  as:

$$\text{coc}(\pi_i^e, \pi_j^e) = 1 - \mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j), \quad (8)$$

where  $\mathbf{c}_i$  and  $\mathbf{c}_j$  are the RGB representation of the colors of the parts of the primitives  $\pi_i^e$  and  $\pi_j^e$  that face each other; and,  $\mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j)$  is Euclidean distance between RGB values of the colors  $\mathbf{c}_i$  and  $\mathbf{c}_j$ .

Co-colority between an edge primitive  $\pi^e$  and a mono primitive  $\pi^m$ , and between two monos can be defined similarly (not shown here).

In Fig. 6, a pair of co-color and not co-color primitives are shown.

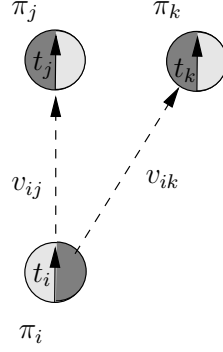


Figure 5: Linear dependence of three  $\pi_i^e$ ,  $\pi_j^e$  and  $\pi_k^e$ . In this example,  $\pi_i^e$  is linearly dependent with  $\pi_j^e$  whereas  $\pi_k^e$  is linearly independent of other primitives.

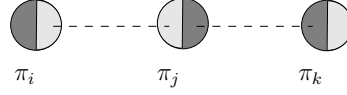


Figure 6: Co-colority of three 2D primitives  $\pi_i^e$ ,  $\pi_j^e$  and  $\pi_k^e$ . In this example,  $\pi_i^e$  and  $\pi_j^e$  are cocolor, so are  $\pi_i^e$  and  $\pi_k^e$ ; however,  $\pi_j^e$  and  $\pi_k^e$  are not cocolor.

## 4 Formulation of the model

For the prediction of the depth at monos, we developed a voting model. In a voting model, there are a set of voters that state their *opinion* about a certain event  $e$ . A voting model combines these votes in a reasonable way to make a decision about the event  $e$ .

In the depth prediction problem, the event  $e$  to be voted about is the depth and the 3D orientation of a mono  $\pi^m$ , and the voters are the edge primitives  $\{\pi_i^e\}$  (for  $i = 1, \dots, N_E$ ) that bound the mono. In this paper, we are interested in the predictions of pairs of  $\pi_i^e$ s, which are denoted by  $P_j$  for  $j = 1, \dots, N_P$ . While forming a pair  $P_j$  from two edges  $\pi_i^e$  and  $\pi_k^e$  from the set of the bounding edges of a mono  $\pi^m$ , we have the following restrictions:

1.  $\pi_i^e$  and  $\pi_k^e$  should share the same color with the mono  $\pi^m$  (i.e., the following relations should hold:  $coc(\pi_i^e, \pi_k^e)$  and  $coc(\pi_i^e, \pi^m)$ ).
2. The 3D primitives  $\Pi_i^e$  and  $\Pi_k^e$  of  $\pi_i^e$  and  $\pi_k^e$  should be on the same plane (i.e.,  $cop(\Pi_i^e, \Pi_k^e)$ ).
3.  $\pi_i^e$  and  $\pi_k^e$  should not be linearly dependent so that they can define only one plane (i.e.,  $\neg lin(\pi_i^e, \pi_k^e)$ ).

In figure 7, such restrictions are illustrated for an example mono and a set of edge primitives that bound it. The primitives  $\pi_j^e$  and  $\pi_m^e$  are on the same line (i.e., they are linearly dependent), and they define infinitely many planes. As for primitives  $\pi_l^e$  and  $\pi_k^e$ , they cannot define a plane as they are not on the same plane, nor do they share the same color.

The vote  $v_i$  by a pair  $P_j$  can be parametrized by:

$$v_i = (\mathbf{X}, \mathbf{n}), \quad (9)$$

where  $\vec{n}$  is the normal of the mono  $\pi^m$ , and  $z$  is its depth relative to the plane defined by  $P_i$ .

Each  $v_i$  has an associated reliability or probability  $r_i$ . They denote how likely the vote is based on the believes of pair  $P_i$ . It can be modeled as a function of the distance of the mono  $\pi^m$  to the intersection point

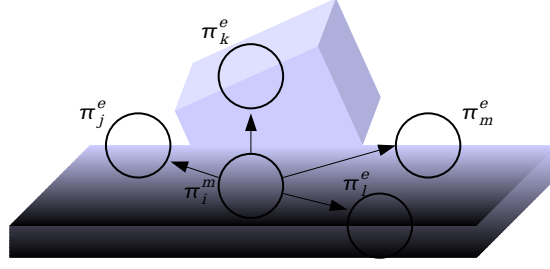


Figure 7: A set of primitives for illustrating why the relations coplanarity, cocolority and linear dependence are required as restrictions for forming pairs from edges.

IP:

$$r_i = f(d(\Pi^m, P_i)). \quad (10)$$

$r_i$  can be weighted by the confidences of the elements of the pair  $P_i$  that reflect their quality.

#### 4.1 Bounding edges of a mono

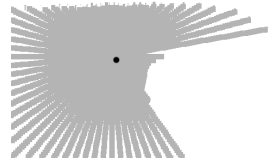
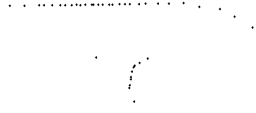





Search Area	Without Grouping	With Grouping	Input Image
a) 			
b) 			

Figure 8: Finding bounding edge primitives with and without grouping information for two different monos which are marked in black in the first column. Using grouping information produces a more complete boundary finding as shown in (a). However, using grouping may include unwanted edge primitives in the boundary as shown in (b).

Finding the bounding edges of a mono  $\pi^m$  requires making searches in a set of directions  $d_i, i = 1 \dots N_d$  for the edge primitives. In each direction  $d_i$ , starting from a minimum distance  $R_{min}$ , the search is performed upto a distance of  $R_{max}$  in discrete steps  $s_j, j = 1 \dots N_s$ . If an edge primitive  $\pi^e$  is found in direction  $d_i$  in the neighborhood  $\Omega$  of a step  $s_j$ ,  $\pi^e$  is added to the list of bounding edges and the search continues with the next direction.

The above mentioned method for finding the bounding edge primitives will lead to an incomplete and sparse boundary detection (see figure 8) because the search is performed only in a set of discrete directions. This can be improved by making use of the contour grouping information; when an edge primitive  $\pi^e$  is found in a direction  $d_i$  at step  $s_j$ , if  $\pi^e$  is part of a group  $G$ , then all the edge primitives in  $G$  can be added to the list of bounding edges (see [23] for information about the grouping method we employ in this paper).

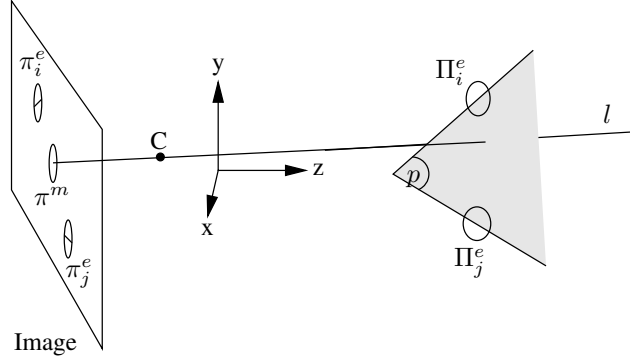


Figure 9: Illustration of how the vote of a pair of edge primitives is computed. The 3D primitives  $\Pi_i^e$  and  $\Pi_j^e$  corresponding to the 2D primitives  $\pi_i^e$  and  $\pi_j^e$  define the plane  $p$ . The intersection of  $p$  with the ray  $l$  that goes through the 2D mono  $\pi^m$  and the camera center  $C$  then determines the position of the estimated 3D mono  $\Pi^m$ . The 3D orientation of  $\Pi^m$  is set to be the orientation of the plane  $p$ .

Grouping information can lead to more complete and dense boundary finding as shown in figure 8(a); however, for certain objects, it may lead to worse results due to low contrast edges (see figure 8(b)).

#### 4.2 The vote of a pair of edge primitives on a mono $\pi^m$

A pair  $P_i$  of two edge primitives  $\pi_j^e$  and  $\pi_k^e$  with two corresponding 3D edge primitives  $\Pi_j^e$  and  $\Pi_k^e$ , which are co-planar, co-color and linearly *independent*, defines a plane  $p$  with 3D normal  $\mathbf{n}$  and position  $\mathbf{X}$ .

The vote  $v_l$  of  $\Pi_j^e$  and  $\Pi_k^e$  is computed by the intersection of the plane  $p$  with the ray  $l$  that goes through the mono,  $\pi^m$ , and the focus of the camera (see figure 9). The ray  $l$  is computed using the following formula ([5], pg41):

$$\mathbf{X}_a = P^{-1}(-\tilde{p} + \lambda \tilde{x}), \quad (11)$$

where  $\tilde{x}$  is the homogeneous position of  $\pi^m$ ;  $P$  and  $\tilde{p}$  are respectively the 3x3 and the 3x1 sub-parts of the 3x4 projection matrix  $P_m$  so that  $P_m = [P \ \tilde{p}]$ ; and,  $\lambda$  is an arbitrary number. By using two different values for  $\lambda$ , two different points on ray  $l$  are extracted which then are used to compute the ray  $l$ .

Because the ray  $l$  is unique for a mono  $\pi^m$ , all the votes processed for the mono  $\pi^m$  will be on ray  $l$ . This property can be exploited for clustering the votes as discussed in section 4.3

#### 4.3 Combining the votes

The votes can be integrated using different ways to estimate the 3D representation  $\Pi^m$  of a 2D mono  $\pi^m$ :

- *Weighted averaging:*

$$\Pi^m = C \sum_{i=1}^{N_P} v_i r_i, \quad (12)$$

where  $C$  is a normalization constant.

- *Clustering:*

Weighted averaging is prone to outliers which can be overcome by utilizing the set of clusters in the

votes. Let us denote the clusters by  $c_i$  for  $i = 1, \dots, N_c$ . Then, one integration scheme would be to take the cluster that has the highest average reliability:

$$\Pi^m = \arg \max_{c_i} \frac{1}{\#c_i} \sum_{v_j \in c_i} r_j. \quad (13)$$

where  $r_i$  is the reliability (*i.e.*, confidence) associated to the vote  $v_i$ .

An alternative can use the most crowded cluster:

$$\Pi^m = \arg \max_{c_i} \#c_i. \quad (14)$$

It is also possible to combine the number of votes and the average reliability of a cluster for making a decision.

As mentioned above, weighted averaging is prone to outliers but is fast. Clustering the votes can filter outliers whereas is slow. Moreover, clustering is an ill-posed problem, and most of the time, it is not trivial to determine the number of clusters from the data points that will be clustered.

In this paper, we implemented (1) a histogram-based clustering where the number of bins is fixed, and the best cluster is considered to be the bin with the most number of elements, and (2) a clustering algorithm where the number of clusters is determined automatically by making use of a cluster-regularity measure and maximizing this measure iteratively.

(1) is a simple but fast approach whereas (2) is considerably slower due to the iterative-clustering step. Suprisingly, our investigations showed that (1) and (2) produce almost identical results (the comparative results are not provided in this paper). For this reason, we have adopted (1) as the clustering method for the rest of the paper.

#### 4.4 Combining the predictions using area information

3D surfaces project as areas into 2D images. Although one surface may project as many areas in the 2D image, it can be claimed that the image points in an image area are part of the same 3D surface[SK: This assumption does not always hold. I need to elaborate.].

Figure 10 shows the predictions of a surface. Due to strong outliers in the stereo computation, depth predictions are scattered around the surface that they are supposed to represent. We show that it is possible to segment the 2D image into areas based on intensity similarity and combine the predictions in areas to get a cleaner and more complete surface prediction.

We segment an input image  $\mathcal{I}$  into areas  $A_i$ ,  $i = 1, \dots, N_A$  using co-colority (see section 3) between primitives utilizing a simple region-growing method; the areas are grown until the image boundary or an edge-like primitive is hit. Figure 11 shows the segmentation of one of the images from figure 1.

In this paper, we assume that each  $A_i$  has a corresponding surface  $S_i$  defined as follows:

$$S_i(x, y, z) = ax^2 + by^2 + cz^2 + dxy + eyz + fxz + gx + hy + iz = 1. \quad (15)$$

Such a surface model allows a wide range of surfaces to be represented, including spherical, ellipsoid, quadratic, hyperbolic, conic, cylinderic and planar surfaces.

$S_i$  is estimated from the predictions in  $A_i$  by solving for the coefficients using a least-squares method. As there are nine coefficients, such a method requires at least nine predictions to be available in area  $A_i$ . For the predictions shown in figure 10, the following surface is estimated which is shown in figure 12 using a sparse sampling (only non-zero coefficients are shown):

$$S_0 = 1.5 \times 10^{-5}y^2 + 5 \times 10^{-6}yz - 1.9 \times 10^{-4}x + 8 \times 10^{-3}y + 1.2 \times 10^3z = 1. \quad (16)$$



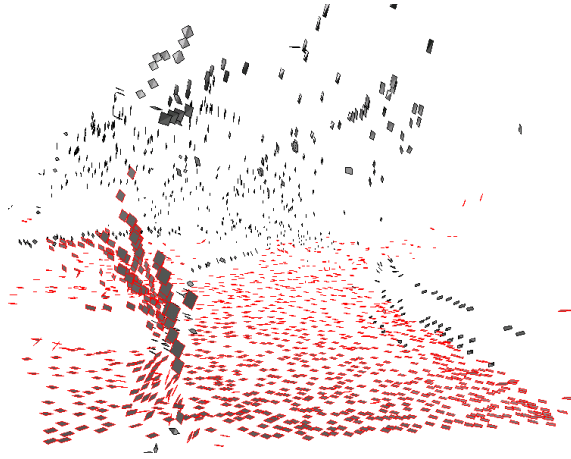


Figure 10: The predictions on the surface of the road for the input images shown in figure 1 (predictions are marked with red boundaries). The predictions are scattered around the plane of the road, and there are wrong predictions due to strong outliers in the computed stereo.

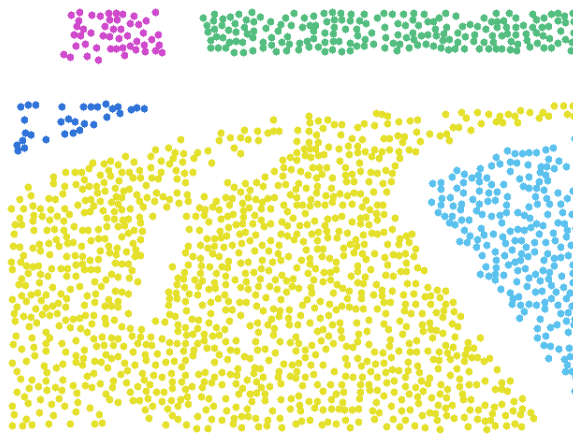


Figure 11: Segmentation of one of the input images given in 1 into areas using region-growing based on primitives.

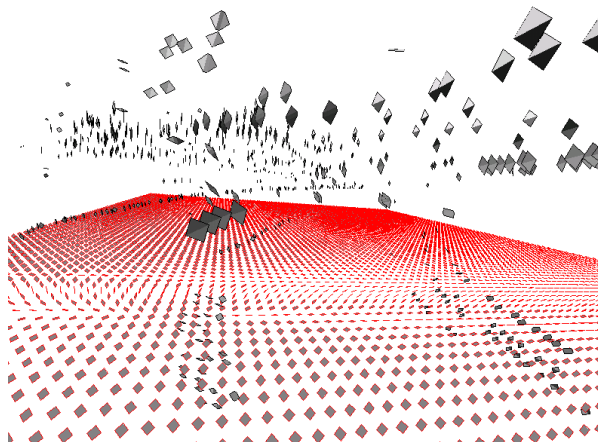


Figure 12: The surface given in equation 16 which is extracted from the predictions shown in figure 10.

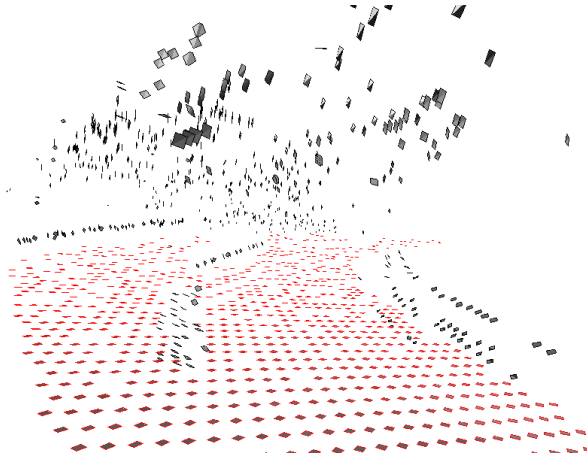


Figure 13: The predictions from 10 that are corrected using the extracted surface  $S_0$  shown in equation 16 and figure 12.

$S_0$  in equation 16 is mainly a planar surface with small quadratic coefficients caused by outliers. Having an estimated  $S_i$  for an area  $A_i$ , it is possible to *correct* the mono predictions using the estimated surface  $S_i$ : Let  $\mathbf{X}_n$  be the intersection of the surface  $S_i$  with the ray that goes through  $\pi^m$  and the camera, and  $\mathbf{n}_n$  be the surface normal at this point (defined by  $\mathbf{n}_n = (\delta S_i / \delta x, \delta S_i / \delta y, \delta S_i / \delta z)$ ).  $\mathbf{X}_n$  and  $\mathbf{n}_n$  are respectively the corrected position and the orientation of mono  $\Pi^m$ . Corrected 3D monos for the example scene is shown in figure 13. Comparison with the initial predictions which are shown in figure 10 concludes that (1) outliers are *corrected* with the extracted surface representation, and (2) orientations and positions are qualitatively better.

## 5 Results

The test cases include kitchen scenarios and road scenarios which are intended for PACO+ and Drivscop projects, respectively. The results of our model is shown for a few examples in figures 14, 15, 16, 17 and 18.

The results show that inspite of limited 3D information from feature-based stereo which may contain strong outliers in some of the scenes (as shown in figure 1), our result is able to predict the surfaces.

## 6 Conclusion

In this paper, we introduced a voting model that estimates the depth at homogeneous or weakly-textured image patches (called monos) from the depth of the bounding edge-like structures. The depth at edge-like structures is computed using a feature-based stereo algorithm [22], and is used to vote for the depth of a mono, which otherwise is not possible to compute easily due to the correspondence problem.

The method presented in this paper is an ongoing work. In the future, the reliability of each vote will be replaced by the statistics collected from chromatic range data (see [12]). Moreover, comprehensive comparison as well as possible combination with dense stereo methods are going to be investigated.

## 7 Acknowledgments

This work is supported by Drivscop projects.

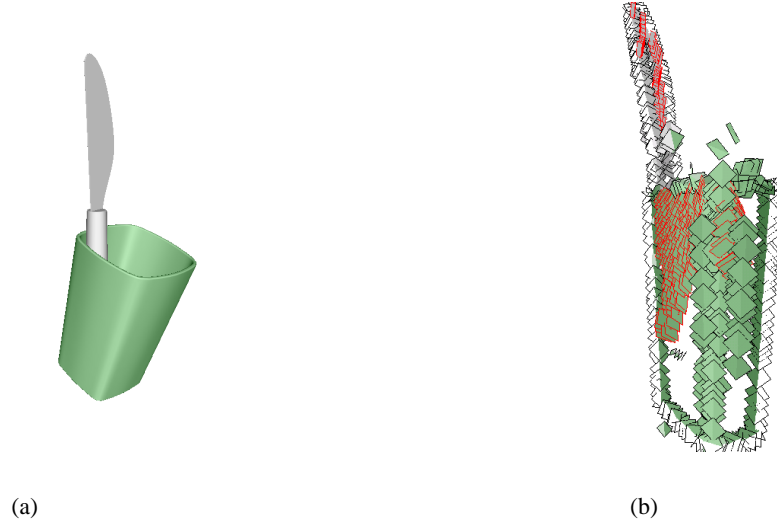


Figure 14: Experiment results on an artificial *kitchen* scene. **(a)** Left image of the input stereo pair. **(b)** The predictions of our model.

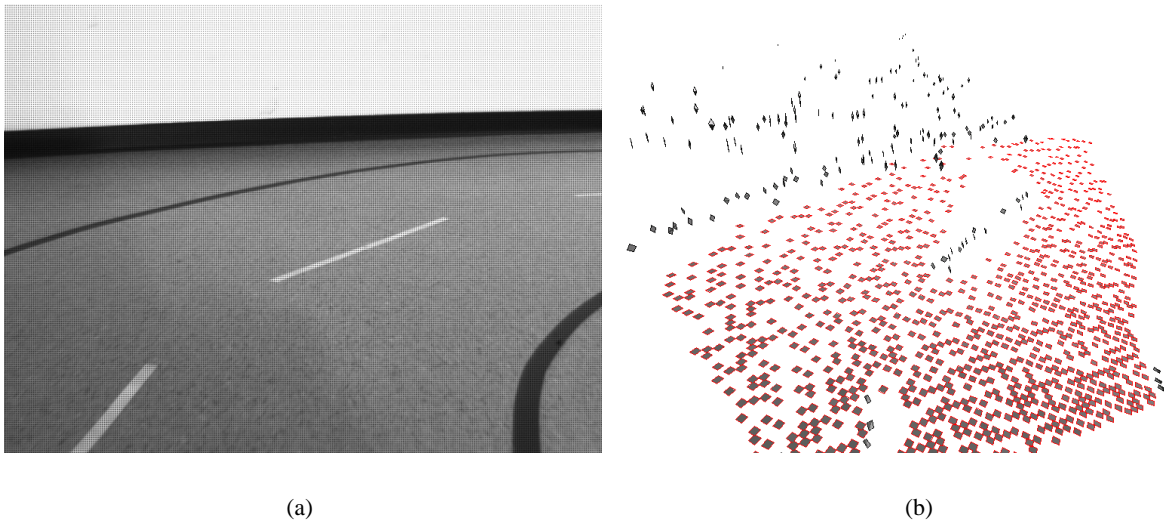
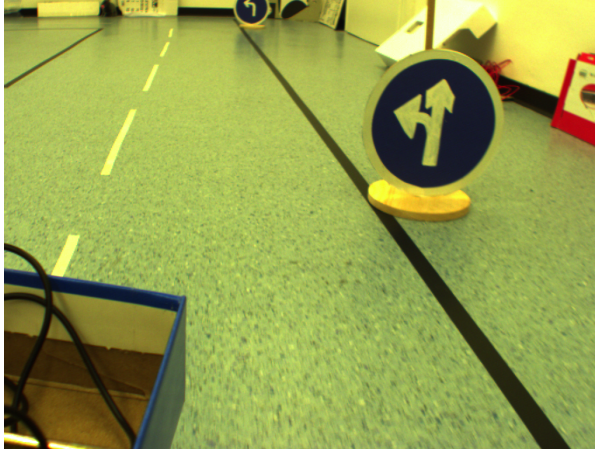
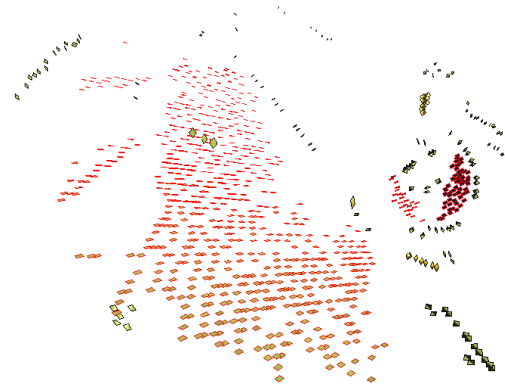


Figure 15: Experiment results on a road scene. **(a)** Left image of the input stereo pair. **(b)** The predictions of our model.



(a)



(b)

Figure 16: Experiment results on a road scene. **(a)** Left image of the input stereo pair. **(b)** The predictions of our model.

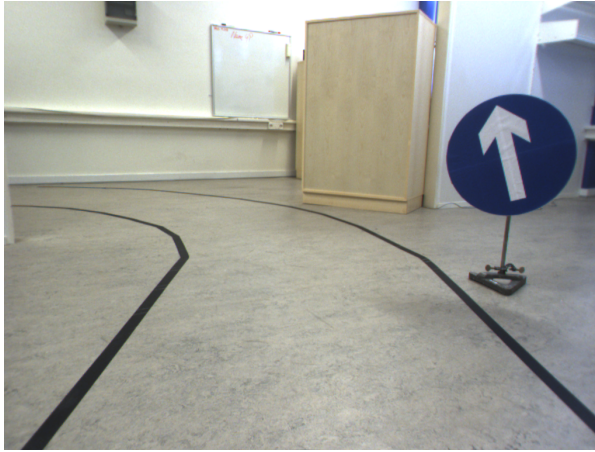


(a)

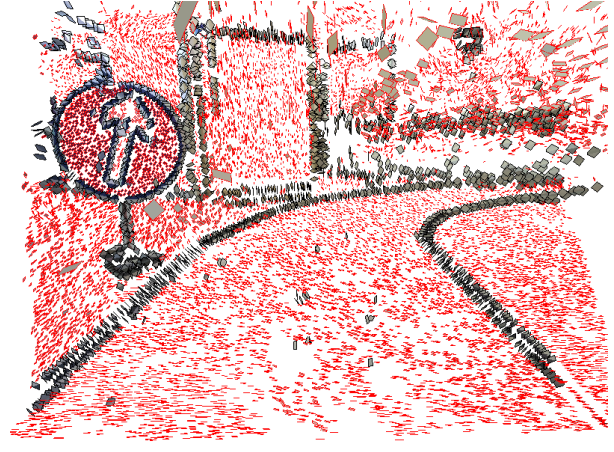


(b)

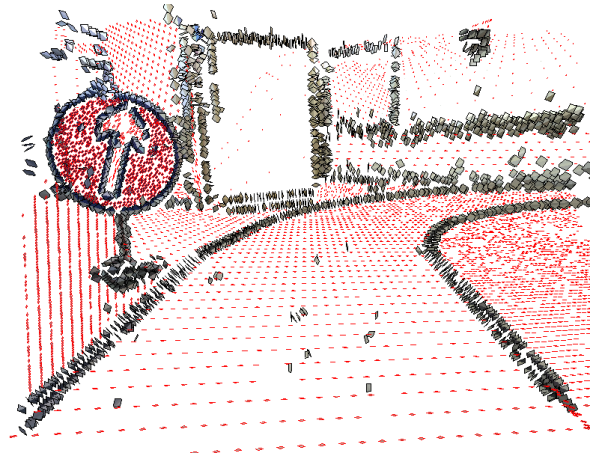
Figure 17: Experiment results on a *kitchen* scene. **(a)** Left image of the input stereo pair. **(b)** The predictions of our model.



(a)



(b)



(c)

Figure 18: Experiment results on an indoor road scene. **(a)** Left image of the input stereo pair. **(b)** The predictions without corrections from the fitted surfaces. **(c)** The predictions after surface corrections. Note that due to outliers in the predictions, surface fitting may not improve original predictions.



## References

- [1] H. G. Barrow and J. M. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116, 1981.
- [2] A. B.L., S. M., and F. R.W. The interpolation of object and surface structure. *Cognitive Psychology*, 44:148–190(43), March 2002.
- [3] V. Bruce, P. R. Green, and M. A. Georgeson. *Visual Perception: Physiology, Psychology and Ecology*. Psychology Press, 4th edition, 2003.
- [4] T. S. Collett. Extrapolating and Interpolating Surfaces in Depth. *Royal Society of London Proceedings Series B*, 224:43–56, Mar. 1985.
- [5] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.
- [6] W. E. L. Grimson. A Computational Theory of Visual Surface Interpolation. *Royal Society of London Philosophical Transactions Series B*, 298:395–427, Sept. 1982.
- [7] W. E. L. Grimson. Binocular shading and visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 28(1):19–43, 1984.
- [8] G. Guy and G. Medioni. Inference of surfaces from sparse 3-d points. In *ARPA94*, pages II:1487–1494, 1994.
- [9] D. Hubel and T. Wiesel. Anatomical demonstration of columns in the monkey striate cortex. *Nature*, 221:747–750, 1969.
- [10] S. Kalkan, N. Pugeault, and N. Krüger. Perceptual operations and relations between 2d or 3d visual entities. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-3, 2007.
- [11] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of local 3d structure in 2d images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1114–1121, 2006.
- [12] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of second-order relations of 3d structures. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.
- [13] K. Kang, J.-P. Tarel, R. Fishman, and B. D. Cooper. A linear dual-space approach to 3D surface reconstruction from occluding contours using algebraic surface. In *International Conference on Computer Vision*, volume 1, pages 198–204, 2001.
- [14] N. Krüger and M. Felsberg. An explicit and compact coding of geometric and structural information applied to stereo matching. *Pattern Recognition Letters*, 25(8):849–863, 2004.
- [15] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5):417–428, 2004.
- [16] N. Krüger, N. Pugeault, and F. Wörgötter. Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-4, 2007.
- [17] N. Krüger, N. Pugeault, and F. Wörgötter. Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information. *To be submitted.*, 2007.
- [18] M. S. Lee and G. Medioni. Inferring segmented surface description from stereo data. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 346, Washington, DC, USA, 1998. IEEE Computer Society.

- [19] M.-S. Lee, G. Medioni, and P. Mordohai. Inference of segmented overlapping surfaces from binocular stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):824–837, 2002.
- [20] X. Liu, H. Yao, and W. Gao. Shape from silhouette outlines using an adaptive dandelion model. *Computer Vision and Image Understanding*, 105(2):121–130, 2007.
- [21] V. S. Nalwa. Line-drawing interpretation: Bilateral symmetry. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(10):1117–1120, 1989.
- [22] N. Pugeault and N. Krüger. Multi-modal matching applied to stereo. *Proceedings of the BMVC 2003*, pages 271–280, 2003.
- [23] N. Pugeault, F. Wörgötter, and N. Krüger. Multi-modal scene reconstruction using perceptual grouping constraints. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR’06)*, 2006.
- [24] K. A. Stevens. The visual interpretations of surface contours. *Artificial Intelligence*, 17:47–73, 1981.
- [25] D. Terzopoulos. Multi-level reconstruction of visual surfaces: Variational principles and finite element representations. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1982.
- [26] D. Terzopoulos. The computation of visible-surface representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4):417–438, 1988.
- [27] F. Ulupinar and R. Nevatia. Constraints for interpretation of line drawings under perspective projection. *CVGIP: Image Underst.*, 53(1):88–96, 1991.
- [28] F. Ulupinar and R. Nevatia. Perception of 3-d surfaces from 2-d contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(1):3–18, 1993.

## **Appendix G**

### ***Multi-modal Primitives (submitted): Local, Condensed, and Semantically Rich Visual Descriptors and the Formalisation of Contextual Information***

N. Krüger, N. Pugeault and F. Wörgötter  
( submitted )



Robotics Group  
The Maersk Mc-Kinney Moller Institute  
University of Southern Denmark

---

Technical Report no. 2007 – 4

---

# **Multi-modal Primitives: Local, Condensed, and Semantically Rich Visual Descriptors and the Formalisation of Contextual Information**

Norbert Krüger, Nicolas Pugeault and Florentin Wörgötter

April 24, 2007

Title	Multi-modal Primitives: Local, Condensed, and Semantically Rich Visual Descriptors and the Formalisation of Contextual Information
	Copyright © 2007 Norbert Krüger, Nicolas Pugeault and Florentin Wörgötter. All rights reserved.
Author(s)	Norbert Krüger, Nicolas Pugeault and Florentin Wörgötter
Publication History	Second version. This contains the submitted paper, plus the deleted parts (appendices, footnotes, etc.).

## Abstract

We present a novel representation of visual information, based on local symbolic descriptors that we call primitives. These primitives: (1) combine different visual modalities, (2) associate semantic to local scene information, (3) reduce the bandwidth of the information exchanged across the system. First, 2D primitives are extracted from images. In a second step, stereo-pairs of 2D-Primitives are used to reconstruct information about the scene structure leading to 3D-Primitives with additional semantic properties.

Since the Primitives allow for strong predictions, based on statistical dependencies as well as the deterministic change of image structure under coherent motion, they serve to initiate a disambiguation process and form a link to higher level cognitive tasks. In this context, we briefly describe different applications of our representation: (1) their role in an early cognitive architecture integrating perceptual grouping and motion (2) depth prediction at homogeneous image patches, (3) learning of object representations, and (4) grasping in the context of vision based robotics.

We also discuss the distinguishing properties of our representation and compare them with other approaches.

## 1 Introduction

There exists a large amount of evidence that the human visual system, in its first cortical stages, processes a number of aspects of visual data (see, *e.g.*, [1,2]). These aspects, in the following called visual modalities, cover, *e.g.*, local orientation [1,3], colour [3], junction structures [4], stereo [5] and optic flow [3]. At the first stage of visual processing (called 'Early Vision' in [6]), these modalities are computed locally for a certain retinal position. At a later stage (called 'Early Cognitive Vision' in [6]), results of such local processing become integrated with the spatial and temporal context. Computer vision has dealt to a large extent with these modalities separately and in many computer vision systems, one or more of the above-mentioned aspects are processed (see, *e.g.*, [7–9]).

An important problem, the human visual system as well as any artificial visual system has to cope with, is the large amount of ambiguity and noise in these low level modalities that is irreducible by local processes only. Reliable actions require a more stable representation of visual features. As a consequence, a disambiguation process that makes use of contextual information is required. In [10] we have described two main regularities in visual data (well recognised in the computer vision community) that support such a disambiguation process: (i) coherent motion of rigid bodies; and (ii) statistical interdependencies underlying most grouping processes [11–13]. These two regularities allow predictions between locally extracted visual events, and verification of the spatio-temporal coherence of transient perceptual hypotheses.

The establishment of such a disambiguation process presupposes communication of temporal and spatial information, requiring the local representation of visual data to comply with the two properties:

**Property 1 Predictability** *The local representation of visual data allows for rich predictions between related visual events — *e.g.*, the change of position and appearance of a local patch under a rigid body motion.*

and

**Property 2 Limited Bandwidth** *The local representation of visual data reduces the dimensionality of the representation of the local signal. This allows for the process to work with limited bandwidth when spatially and temporally distinct visual events become related by predictions.*

These two properties demand for a *condensed* representation of visual information and it is argued in [14] that the need for properties 1 and 2 naturally results in symbolic representations.

In this work, we present a novel kind of scene representation, based on local symbolic descriptors that we call *visual primitives* (see Fig. 1). A primitive combines different visual modalities into one local feature descriptor (see sections 2 and 3), and thus, allows for a condensed representation of the visual scene (satisfying property 2). Furthermore, primitives allow for the formulate of predictions (property 1) using statistical dependencies from grouping and motion. These statistical dependencies bootstrap a disambiguation process that is described in, *e.g.*, [15, 16].

The system we present processes information over multiple stages (for an overview, see Fig. 1), described in the following sections. In section 2, individual modalities are computed by linear and non-linear filtering processes. Section 3 describes the condensation process that extracts *2D-primitives*. In section 4, stereo-pairs of 2D-primitives are used to infer information about the scene structure, in terms of *3D-primitives*. Section 5 briefly describes some applications where this framework was used. In section 6, we discuss the distinguishing properties of our representation and compare them with other methods.

## 2 Analysis of the local signal structure

In section 2.1, we will first describe how we distinguish different kinds of local image structures. The processing of the modalities (*i.e.*, orientation, phase and optic flow) is described in section 2.2 and 2.3. Fig. 1b illustrates the results of the process described herein.

### 2.1 Intrinsic dimension

Different kinds of image structures coexist in natural images: homogeneous image patches, edges, corners, and textures. Furthermore, certain concepts are only meaningful for specific classes of image structures. For example, the concept of orientation is well defined for edges or lines but not for junctions, homogeneous image patches, or most textures. As another example, the concept of position is different for a junction as compared to an edge or an homogeneous image patch — see Fig. 2.1. In homogeneous areas of the image, no particular location can be defined (Fig. 2.1a); therefore, an equidistant sampling is appropriate. For line or edge structures (Fig. 2.1b), position can be defined using energy maxima. However, because of the aperture problem, the energy maximum will span a one-dimensional manifold, and therefore the feature can be localised only up to this manifold. This results in a fundamental ambiguity in the localisation of local edge or line features. By contrast, a junction’s locus can be unambiguously defined by the intersection of the lines (see Fig. 2.1c). Similar considerations are required for other modalities such as colour, optic flow and stereo (see section 2.3).

Hence, before applying concepts such as orientation or position, we need to classify image patches according to their junction-ness, edge-ness or homogeneous-ness. The intrinsic dimension (see, *e.g.*, [19, 20]) is a suitable classifier in this context [18]. Ideal homogeneous image patches have an intrinsic dimension of zero (id0), ideal edges are intrinsically One-dimensional (id1) while junctions and most textures have an intrinsic dimension of two (id2). Going beyond common discrete classification [20, 21], we use a *continuous* formulation [18, 22, 23] that allows for a formulation of reasonable confidences for the different image structure classes. We classify image patches according to the dimension of the subspace that is occupied by the local spectral energy. When looking at the spectral representation of a local image patch (see Fig. 2.1), we see that the spectral energy of an intrinsically zero-dimensional signal is concentrated in the origin (Fig. 2.1a), whilst the energy

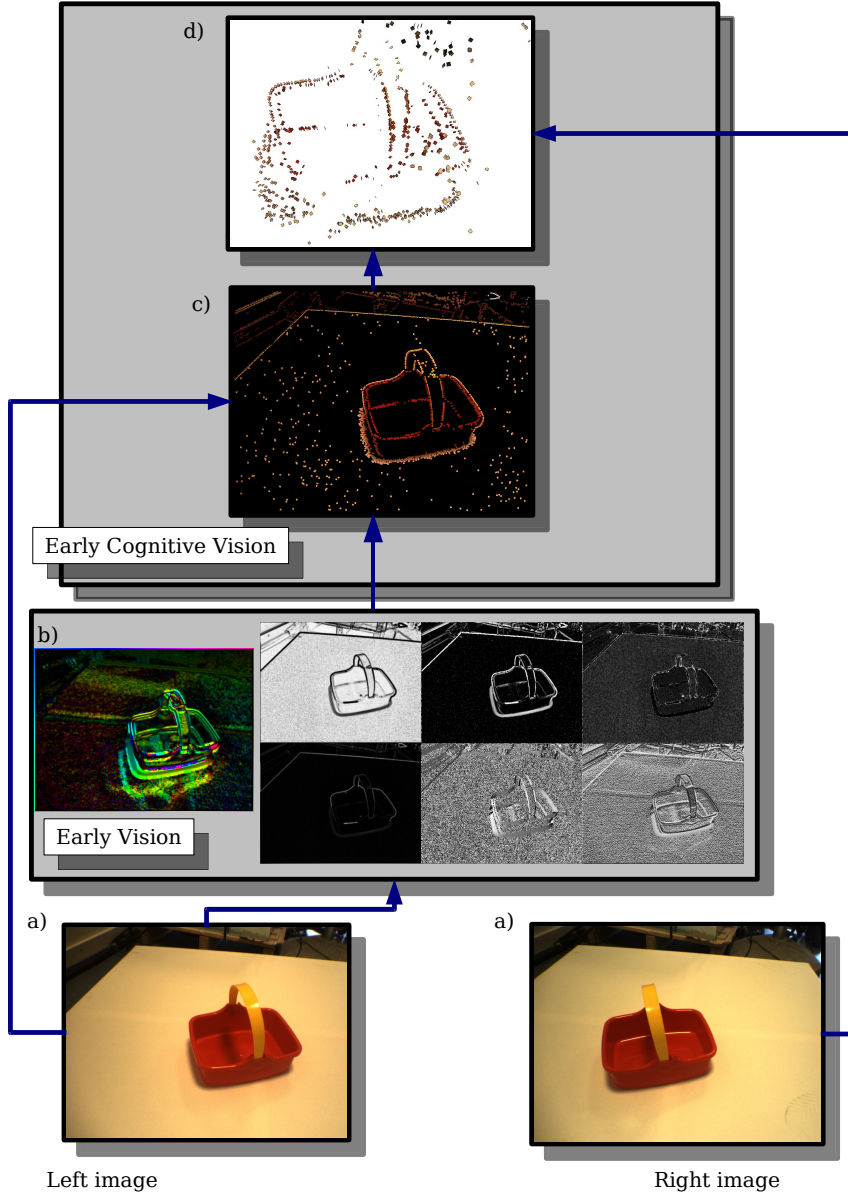


Figure 1: Overview of the primitive extraction scheme. **a)** a stereo-pair of images obtained from a pre-calibrated stereo rig. Therefrom, Early Vision processes are computed as shown in **b)**: the left image shows the optical flow extracted using the Nagel algorithm [17] — see section 2.3. Each pixel represents the local flow at this location by its colour: the hue of indicates the orientation of the flow vector (as shown on the borders of the image) and the intensity the magnitude of the flow (where black stands for a zero flow); the bottom row of images shows the magnitude, orientation and phase of the signal — see section 2.2 — from left to right respectively; The upper row shows the  $id_0$ ,  $id_1$  and  $id_2$  confidences — see section 2.1 — from left to right respectively. In all those graphs the intensity encodes the strength of the filter response (white for high, black for low). In **c)** the information from the early vision is combined in a sparse, condensed way — see section 3. The image shows the primitives extracted from the images shown in **a)** **d)** these primitives are then matched across the two stereo-views and the correspondences thereof allows reconstructing 3D-primitives, that extend naturally the primitive information to 3D space — see section 4.

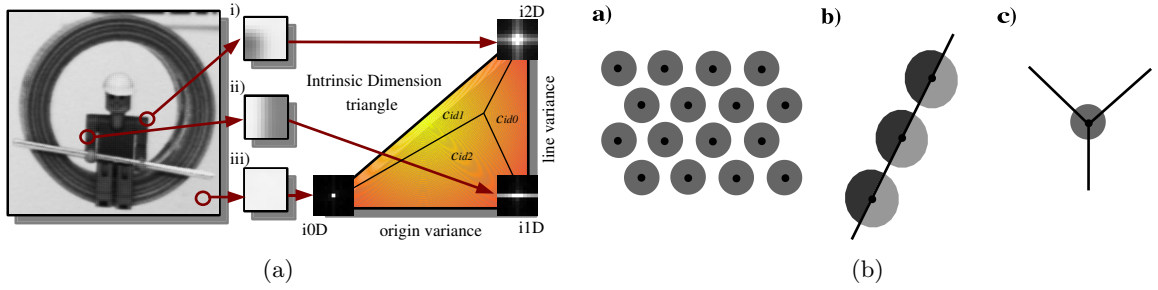


Figure 2: (a) Illustration of the triangular topology of the intrinsic dimension — see [18]; (b) Different localisation problems faced by different classes of image structures: namely a) homogeneous area; b) edge or line; and c) junction (see text).

of an intrinsically one-dimensional signal spans a line (Fig. 2.1b) and the energy of an intrinsically two-dimensional signal varies in more than one dimension (Fig. 2.1c).

Thus, we compute for each pixel position  $\mathbf{x}$ , the three confidences  $c_{id0}(\mathbf{x})$ ,  $c_{id1}(\mathbf{x})$ , and  $c_{id2}(\mathbf{x})$ , that take values in  $[0, 1]$  and add up to one — illustrated, for different scales, in the three bottom rows of Fig. 3. For details of the computation, we refer to [18, 22, 23], and to [24, 25] for some applications of this concept.

The current version of our system focuses on intrinsically one dimensional signals and uses the triangular representation defined above to discard non-edge/non-line structures. There is some ongoing work on the integration of homogeneous (id0) and corner structures (id2) into this framework — see, [25, 26].

## 2.2 Orientation and phase

The extraction of a primitive starts with a rotation invariant quadrature filter that performs a *split of identity* of the signal [27]: it decomposes an intrinsically one-dimensional signal (as defined in the previous section) into local amplitude (see Fig. 3, top row), orientation (see Fig. 3, second row), and phase (see Fig. 3, third row) information.<sup>1</sup>

The local amplitude is an indicator of the likelihood for the presence of an image structure. Orientation encodes the geometric information of the local signal while phase can be used to differentiate between different image structures ignoring orientation differences. Phase for possible grey level structures forms a continuum between  $[-\pi, \pi)$  and encodes the grey level transition of the local image patch across the edge (as defined by the orientation) in a compact way (as one parameter only), *e.g.*, a pixel positioned on a bright line on a dark background has a phase of 0 whereas a pixel positioned on a bright/dark edge has a phase of  $-\pi/2$  (see Fig. 4a and, *e.g.*, [27, 29, 30]). Note that phase is  $2\pi$ -periodic and continuous such that a phase of  $-\pi$  represents the same contrast transition as a phase of  $\pi$ . Orientation  $\theta$  (taking values in the interval  $[0, \pi)$ ) and phase  $\varphi$  are topologically organised on a half torus (see Fig. 4c), and if we extend the concept of orientation to that of a direction (therefore taking values in  $[-\pi, \pi)$ , see also [21]) then the topology of the direction/phase space becomes a complete torus (see Fig. 4b). On a local level, the direction is not decidable [29]; therefore, we will use the half torus topology.

This topology is crucial for the definition of suitable metrics for phase and orientation. For example,

<sup>1</sup>Note that amplitude, orientation and phase can be analogously computed by Gabor wavelets or steerable filters and that our representation does not depend on the filter introduced in [27]. For a discussion of different approaches to define harmonic filters as well as their advantages and problems, we refer to [28].

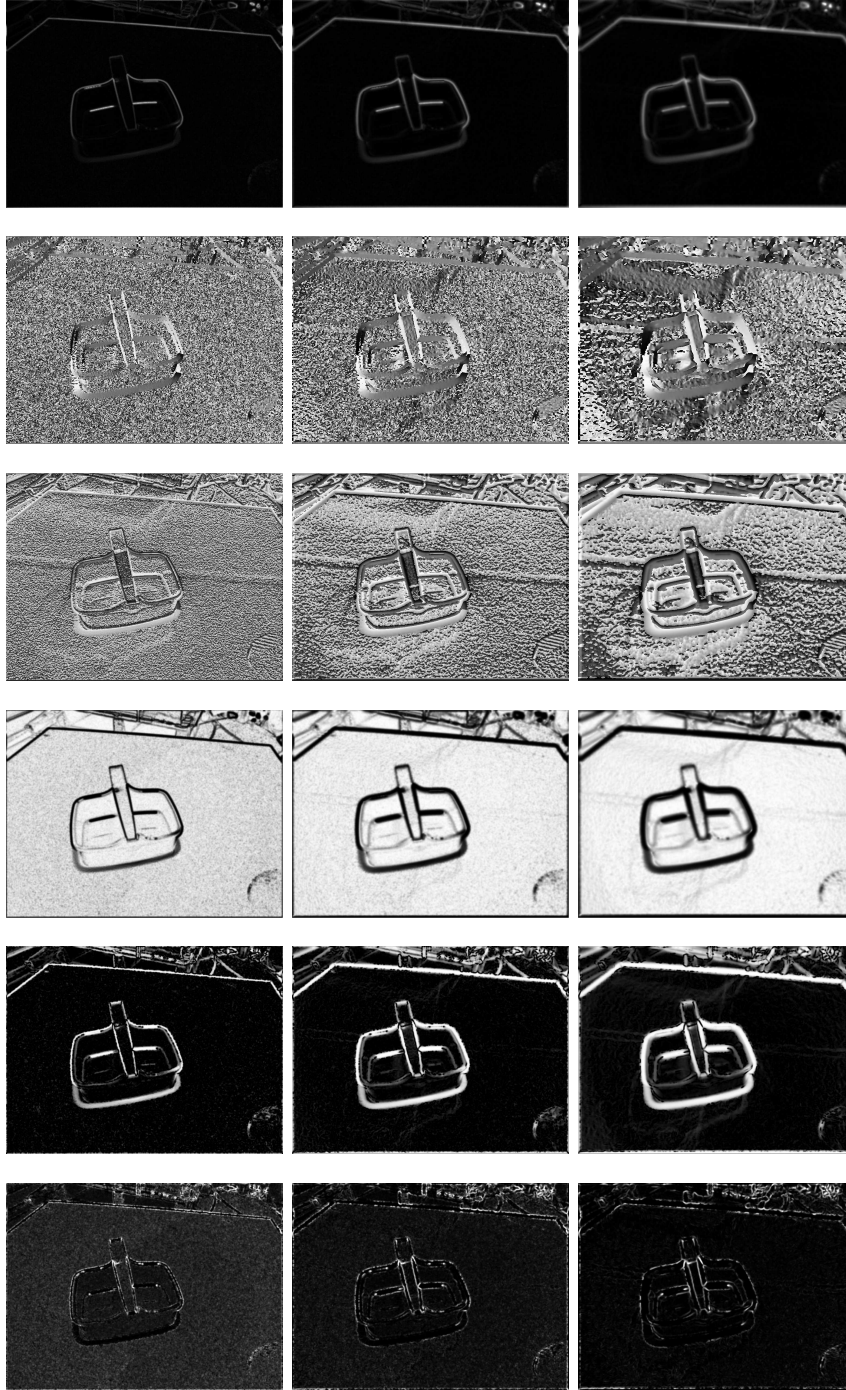


Figure 3: Illustration of the low-level processing for primitive extraction. Each column shows the filter response for a different peak frequency: respectively 0.110 (left), 0.055 (middle) and 0.027 (right). Each row shows response maps for, from top to bottom, local amplitude, orientation, phase, intrinsically zero-Dimensional (id0), one-Dimensional (id1) and two-Dimensional (id2) confidences. In all of those graphs, white stands for a high response and black for a low one.

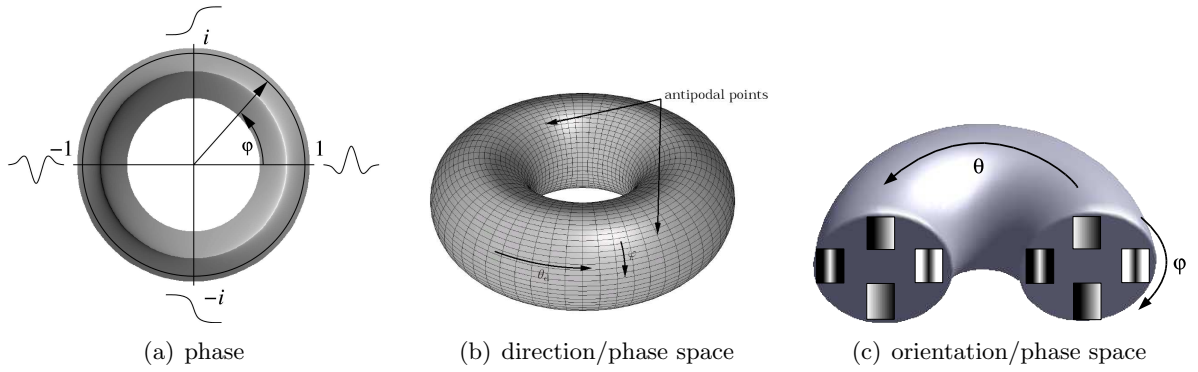


Figure 4: (a) Phase  $\varphi$  describes different intensity transitions, *e.g.*,  $\varphi = \pi$  encodes a dark line on a bright background,  $\varphi = -\pi/2$  encodes a bright–dark edge,  $\varphi = 0$  encodes a bright line on a dark background and  $\varphi = \pi/2$  encodes a dark–bright edge. The phase parameter embeds these distinct cases into a  $2\pi$ –periodic continuum shown in (a). [Acknowledgement: Michael Felsberg] (b) The torus topology of the orientation–phase space. The phase value  $\varphi$  is mapped on the cross section of the torus’ tube whereas the orientation  $\theta$  maps to the revolution angle the torus. (c) When direction is neglected, we get a half torus connected as indicated.

a black–white step edge ( $\varphi = \pi/2$ ) with orientation  $\theta$  should have a small metrical distance to a white–black step edge ( $\varphi = -\pi/2$ ) of orientation  $\pi - \theta$  but a large distance to a black–white step edge of orientation  $\pi - \theta$ . However, a white line on a black background with an orientation  $\theta$  ( $\varphi = 0$ ) should have only a small distance to a white line on a black background with an orientation  $\pi - \theta$  but a large one to any black line on a white background. Therefore, the extremities of the half–torus are linked in a continuous manner as shown in Fig. 4c. For a discussion of the orientation/phase metric, we refer to [31].

Note that there are also some problems involved with filters realising the monogenic signal we are using, as discussed in [28]. First, it turned out that for the monogenic signal it is more difficult to construct filter which allow for stable orientation and phase estimates at high frequencies (compared to, *e.g.*, Gabor wavelets) Second, in the monogenic filter approach there is only one orientation estimate and one phase (in connection to the one orientation) estimate. However, for intrinsically two dimensional signals such as corners and most textures more parameters are needed to represent the local structure (*e.g.*, most textures are characterised by multiple orientations at different frequencies). Third, estimates for, *e.g.*, optic flow can profit from averaging processes over estimates over different orientations. However, in the context of intrinsically one dimensional structures the monogenic signal allows for a good representation.

The application of such a spherical quadrature filter for the processing of our primitives has two main advantages:

1. It allows us to use general advantages of the analytic signal (the aforementioned split of identity, see [29]). Hence, phase is an immediate output of the spherical quadrature filter processing and can directly be used as an attribute that describes the structural information of an oriented image structure (see Fig. 4a).
2. Compared to the use of a Gabor wavelet transform (see, *e.g.*, [?]), we do not need to sample across different orientations: orientation is a direct output of the computation. Hence, we only need to apply 3 filter operations compared to, *e.g.*, 16 for Gabor wavelets (see, *e.g.*, [9]).

We compute filter responses for three different scales, indicated hereafter by the peak frequency of



the associated filter operations.<sup>2</sup> Fig. 3 shows the filter responses in terms of the local amplitude  $m(\mathbf{x})$ , orientation  $\theta(\mathbf{x})$  and phase  $\varphi(\mathbf{x})$ , alongside the resulting primitives, for three scales.

### 2.3 Optic flow and colour

Besides orientation, phase and the intrinsic dimensionality confidences, colour and local optic flow are also associated to the primitive description vector. Kalkan and colleagues [24] compared optic flow algorithms performance depending on the intrinsic dimensionality, *i.e.*, the effect of the aperture problem and the quality on low contrast structures. It appears that different optic flow algorithms are optimal in different contexts. In our system, we primarily use the Nagel–Enkelmann algorithm [33] since it gives stable estimates of the normal flow at id1 structures. We denote the optic flow computed at a position  $\mathbf{x}$  by  $\mathbf{f}(\mathbf{x})$ .

Colour is not processed by filtering operations but sampled (i) on each side of a step edge, or (ii) on each side of a line and on the line itself, depending if the phase describes a step edge or a line structure.

## 3 Condensation scheme

Based on the pixel–wise processing described in section 2, we now want to extract a condensed interpretation of a local image patch by selecting a sparse set of points to which visual modalities become associated. An important aspect of the condensation scheme is that all main parameters can be derived from one property of the basic filter operations called *line/edge bifurcation distance*.

**Definition 1** *The line/edge bifurcation distance  $d_{leb}$  for a given scale is the minimal distance between two edges for them to produce two distinct amplitude maxima.*

Hence, a double edge will be represented by a pair of edge primitives if its width is larger than  $d_{leb}$ , by only one line primitive otherwise. Fig. 5a shows a narrow triangle for which two edges get closer until they meet. Vertical sections of the local local amplitude (Fig. 5b) close to the vertex have only one maximum, that splits into two distinct maxima further away from the vertex, where the distance between the two edges is larger.

Using definition 1 we propose a condensation procedure in three steps: 1) Sampling: the positions of features are computed with sub–pixel accuracy, according to the local intrinsic structure (section 3.1); 2) Elimination: positions that are too close to each other (and therefore would lead to redundant descriptors) are disregarded (section 3.2); 3) Local interpretation: semantic attributes become associated to the computed positions (section 3.3).

Fig. 5c,d, and e, show the primitives extracted after condensation for the three scales used in the present paper — for peak frequencies of 0.11, 0.055 and 0.027, respectively.

### 3.1 Sampling

In section 2.1, it was discussed that the concept of position is different for different type of image structures as defined by the three classes of intrinsic dimensionality. The coding of intrinsic dimension by three values ( $c_{id0}(\mathbf{x})$ ,  $c_{id1}(\mathbf{x})$ ,  $c_{id2}(\mathbf{x})$ ) allows us to select the most likely structure for this patch, and thence to define an appropriate (according to its intrinsic dimension interpretation) position candidate. However, if we do not want to make a decision about the type of local image

<sup>2</sup>Note that step edges have high amplitudes across scales, whilst line structures are represented as a line at coarse scales, and as two step–edges at fine scales, (see section 3 and [32]).

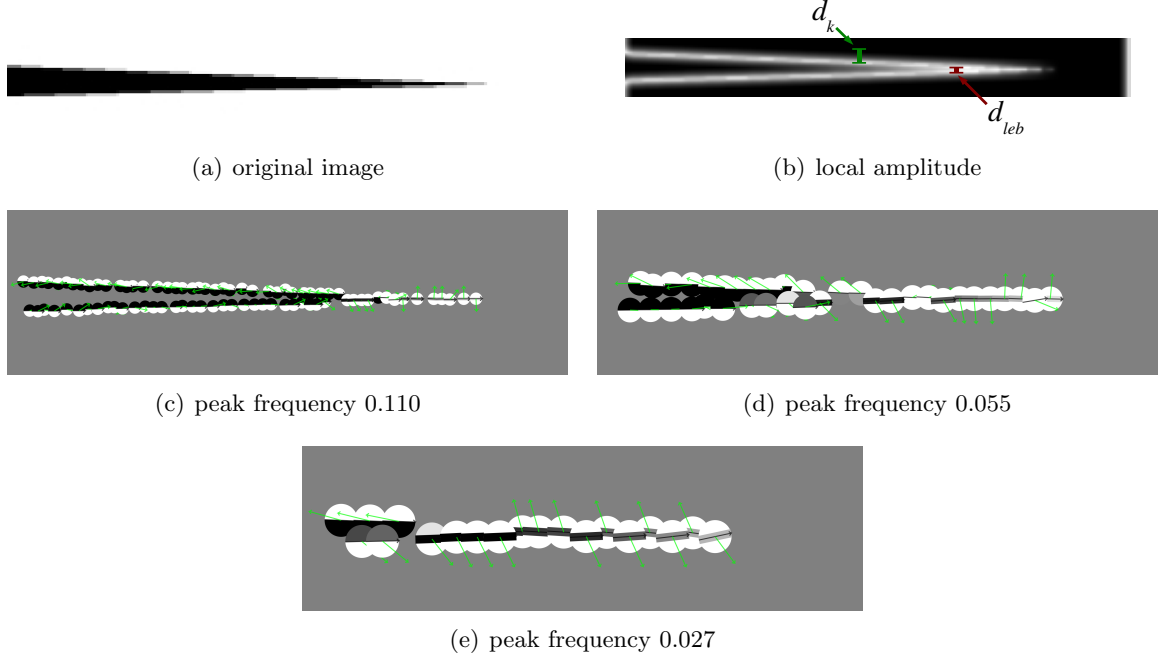


Figure 5: Definition of the elimination parameters  $d_{leb}$  and  $d_k$ . See text.

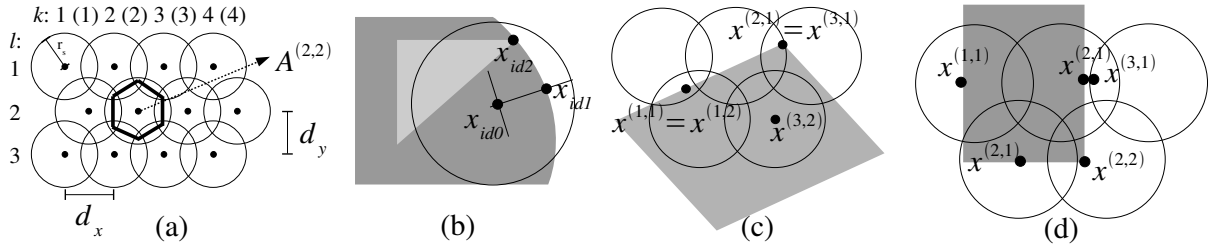


Figure 6: (a) Hexagonal sampling: each hexagon is embedded in a disk  $A(i,j)$ , with a radius  $r_s$ . (b) Three possible hypotheses for positions according to the three different intrinsic dimensions (see section 3.1). (c) Because the disks  $A^{(k,l)}$  overlap, the same position can be found in areas with different index. For these redundant structures, one sample needs to be deleted (see section 3.2.1). (d) Since the local amplitude can still be high for pixels with a certain distance from high contrast structure, an elicited position  $x^{pq}$  may not lie on the edge structure. These positions are redundant since the structure that induced them is already more accurately represented (in terms of position) by other primitives. Therefore, these positions are also disregarded (see section 3.2.2).

Table 1: The scale-dependent parameters of our representation.

Peak frequency (pixels <sup>-1</sup> )	$f_p$	0.1103	0.0551	0.0275
Wavelength (pixels)	$\lambda$	9.06	18.12	36.25
Number of tabs	$n_t$	11	23	33
Line/edge bifurcation (pixels)	$d_{leb}$	3	6	7.5
Hex. grid spacing in $x$ (pixels)	$d_x = 0.85d_{leb}$	2.55	5.1	6.37
Hex. grid spacing in $y$ (pixels)	$d_y = \sqrt{3}/2d_x$	2.21	4.42	5.52
Influence radius (pixels)	$d_k = 2.2d_{leb}$	6.6	13.2	16.5
Condensation rate	$d_{co}$	85%	94%	97%

structure at such an early stage we can also code the three different candidates according to their intrinsic dimension class (see Fig. 6b). These two approaches are implemented by two different modes of the condensation algorithm with different advantages and disadvantages (see below).

To get candidates for our primitives, we first perform a hexagonal sampling (see Fig. 6a) of the image into overlapping areas  $A^{(k,l)}$  with radius  $r_s$ , with  $k, l$  coding the hexagonal grid points. Hexagonal sampling has a number of advantages discussed for example in [34,35]. In the context of this paper, the most important difference with rectangular sampling is that the distance between the centres of neighbour tiles is uniform in an hexagonal grid while in a rectangular grid diagonal spacing is  $\sqrt{2}$  times longer than horizontal or vertical. Since we want to extract symbolic descriptors for each tile, the hexagonal sampling allows for a more evenly distributed symbolic description and reflects more closely the isotropic structure of the original image filters. The parameters  $d_x$  and  $d_y = \frac{\sqrt{3}}{2}d_x$  determine the spatial distance in  $x$  and  $y$  between the centre  $A_c^{(k,l)}$  of the tile  $A^{(k,l)}$  and the centres of the neighbour tiles.<sup>3</sup> For a description of the mathematics of hexagonal sampling we refer to, *e.g.*, [34].

The optimal sampling distance  $d_x$  is related to the line/edge bifurcation distance  $d_{leb}$  — see Fig. 5c, d and e. It turned out that a reasonable estimate for  $d_{leb}$  is:

$$d_{leb} = \frac{1}{3f_p}, \quad (1)$$

hence, we set  $d_x = \text{round}(d_{leb}) + 1$  to be the smallest possible sampling distance within which structures based on the amplitude information can be resolved. Because the line/edge bifurcation distance  $d_{leb}$  depends on the peak frequency  $f_p$ ,<sup>4</sup> so does the sampling distance. All frequency dependent parameters are shown in table 1:

We search on a disk around each  $A_c^{(k,l)}$  for candidate primitives positions. The radius  $r_s$  of this disk is chosen such that each point of the image is covered by at least one of the disks. In a hexagonal grid, the maximum distance to a tile's border is  $\frac{2}{\sqrt{3}}d_x$  hence we set

$$r_s = \text{round}\left(\frac{2}{\sqrt{3}}d_x\right) + 1 \quad (2)$$

We then look for optimal structure dependent primitive positions inside each tile, distinguishing between the three intrinsic dimension's classes:

<sup>3</sup>Note that the odd rows have an onset of  $d_x/2$

<sup>4</sup>note that it is also related to the spatial size, the filter's band-width  $B$ , and the minimal number of tabs  $n_t$ , needed to represent the filter, for a detailed discussion see [28].

### 3.1.1 Homogeneous image patches (id0)

At homogeneous image patches, the position cannot be defined by properties of the local signal since it is constant. Therefore, the position  $\mathbf{x}_{id0}^{(k,l)}$  of a primitive representing an image patch  $A^{(k,l)}$  is defined by equidistant sampling (see Fig. 2.1a):

$$\mathbf{x}_{id0}^{(k,l)} = A_c^{(k,l)}. \quad (3)$$

This is illustrated in Fig. 7b.

### 3.1.2 Lines and edges (id1)

For a line or edge, the position  $\mathbf{x}_{id1}^{(k,l)}$  can be defined through energy maxima that are organised as a one-dimensional manifold. Therefore, an equidistant sampling along these energy maxima is appropriate (see Fig. 2.1b). For this, we look within the area  $A^{(k,l)}$  for the energy maximum along a line orthogonal to the orientation at  $A_c^{(k,l)}$ :

$$\mathbf{x}_{id1}^{(k,l)} = \max_{\mathbf{x} \in g^{(k,l)}} m(\mathbf{x}), \quad (4)$$

where  $g^{(k,l)}$  is a local line going through  $A_c^{(k,l)}$  with orientation perpendicular to  $\theta(A_c^{(k,l)})$ . This is illustrated in Fig. 7c.

### 3.1.3 Junction-like structures (id2)

For a junction, the position  $\mathbf{x}_{id2}^{(k,l)}$  can be defined unambiguously as the maximum of the i2D confidence in a local region (see Fig. 2.1c and [27]):

$$\mathbf{x}_{id2}^{(k,l)} = \max_{A^{(k,l)}} \{c_{id1}(\mathbf{x})\}. \quad (5)$$

This is illustrated in Fig. 7d. <sup>5</sup>

Our system runs in two modes. In the first mode, hereafter named *complete mode*, all three hypotheses are conserved (see Fig. 6b). However, the position corresponding to the maximum of three confidences ( $c_{id0}(\mathbf{x}), c_{id1}(\mathbf{x}), c_{id2}(\mathbf{x})$ ) is called the *reference position*  $\mathbf{x}^{(k,l)}$ , and is used thereafter, in the reduction of redundant descriptors, to compete with proximate candidates. In the second mode, named *contour mode*, we only look at intrinsically one-dimensional signals, *i.e.*, we do the positioning according to Fig. 2.1b. The first mode allows for a complete representation of the signal by also taking into account id0 and id2 structures. However, symbolic representation and 3D reconstruction of id0 and id2 signals are ongoing research topics (see, *e.g.* [25,26]). In the second mode, the primitives symbolic representation, 3D reconstruction (see section 4), and structural relations (such as co-colourity, symmetry and co-planarity), are well defined (see section 5.1).

All positions are computed with sub-pixel accuracy using the formula:

$$\begin{aligned} \tilde{x}_0 &= \frac{1}{s_g} \sum_{i=-w_s}^{w_s} \sum_{j=-w_s}^{w_s} m(x_0 + i, y_0 + j)(x_0 + i), \\ \tilde{y}_0 &= \frac{1}{s_g} \sum_{i=-w_s}^{w_s} \sum_{j=-w_s}^{w_s} m(x_0 + i, y_0 + j)(y_0 + j), \end{aligned} \quad (6)$$

---

<sup>5</sup>Note however, that it is well known that for energy based junction detectors there is a systematic bias towards the inside of the junction (see, *e.g.*, [?]). In [?, 18], we show that by making use of model knowledge (*i.e.*, understanding junctions as points in which lines intersect), a more precise localisation can be ensured.

with  $m(x, y)$  being the local amplitude at pixel position  $(x, y)$  and

$$s_g = \frac{1}{\sum_{i=-w_s}^{w_s} \sum_{j=-w_s}^{w_s} m(x_0 + i, y_0 + j)}, \quad (7)$$

where  $w_s$  is set to  $w_s = d_{leb}$ . In section 3.3, the extracted features phase and orientation are computed at the sub-pixel position, using bi-linear interpolation. Fig. 7b,c,d, show the positions found for different intrinsic dimensions; Fig. 7e,f,g, show the primitives for those locations; Fig. 7h,i,j, show the primitives extracted, in contour mode, with an origin variance  $> 0.3$  and a line variance  $< 0.3$  are shown for the three scales considered in this work: namely for peak frequencies of 0.110 (Fig. 7h), 0.055 (Fig. 7i), and 0.027 (Fig. 7j). Different scales highlight different structures in the scene. Furthermore, a lower peak frequency removes image noise and generates less spurious primitives, whilst smaller structure of the image is become neglected — see [32,36] for a discussion of the effect of scale in edge detection.

We evaluated the accuracy of the primitive extraction on a synthetic image pair, featuring a red circle on black background, and recorded the results in Fig. 8. The top images compare the primitives extracted with (left) and without (right) the sub-pixel localisation of the primitives. Note that the sub-pixel localisation implicitly assumes a symbolic interpretation of the primitive since it associates a meaning to a position (see also the discussion in section 6). Hence, we mainly consider id1 primitives in the following. Effectively we only considered primitives with an origin variance larger than 0.3 and a line variance lower than 0.3. The top images in Fig. 8a show the 2D-primitives extracted and the bottom ones show the 3D-primitives reconstructed using stereopsis. The 3D-primitives are shown from front and side views to illustrate the quality of the depth reconstruction. It is visible in these graphs that the accuracy of 3D reconstruction is greatly improved by a sub pixel localisation of the primitives. The accuracy of the reconstruction decreases towards horizontal primitives due to the inaccuracy of stereo-matching on lines parallel to the epipolar geometry.

Different levels of Gaussian noise were applied to the images, and the accuracy of the extracted primitives were recorded in the graphs 8 (b), (c), (d) and (e). The solid lines show values with sub-pixel accuracy and the dashed ones without. In graph (b) the density of the primitives extracted depending on the noise is shown. As noise tends to increase the line variance in an image patch, less id1D primitives become extracted with larger noise levels. The next graphs chart the localisation (c), orientation (d) and phase (e) errors for different noise levels. As a summary these results show that sub-pixel localisation provides significantly better accuracy, both for 2D-primitives and for 3D reconstruction.

### 3.2 Elimination of redundant descriptors

Since areas  $A^{(k,l)}$  are overlapping, the process described above can lead to identical positions found in neighbouring areas: in Fig. 6c, the putative positions  $\mathbf{x}^{(2,1)}$  and  $\mathbf{x}^{(3,1)}$ , elicited by two distinct hexagonal cell, represent the same image location. Moreover, the filters spatial extension can lead to proximate positions describing essentially the same image structure (see Fig. 6d,  $\mathbf{x}^{(2,1)}$  and  $\mathbf{x}^{(3,1)}$ ).

Therefore we apply an additional process where these redundant descriptors become eliminated. This elimination process faces the following challenges:

- Proximate, yet distinct, putative positions should be preserved. For example, in the triangle in Fig. 5 two edges converge. At some point, these edges become interpreted as a line and the position should be on this line and the phase should become 0 or  $\pm\pi$ . Until then, the triangle

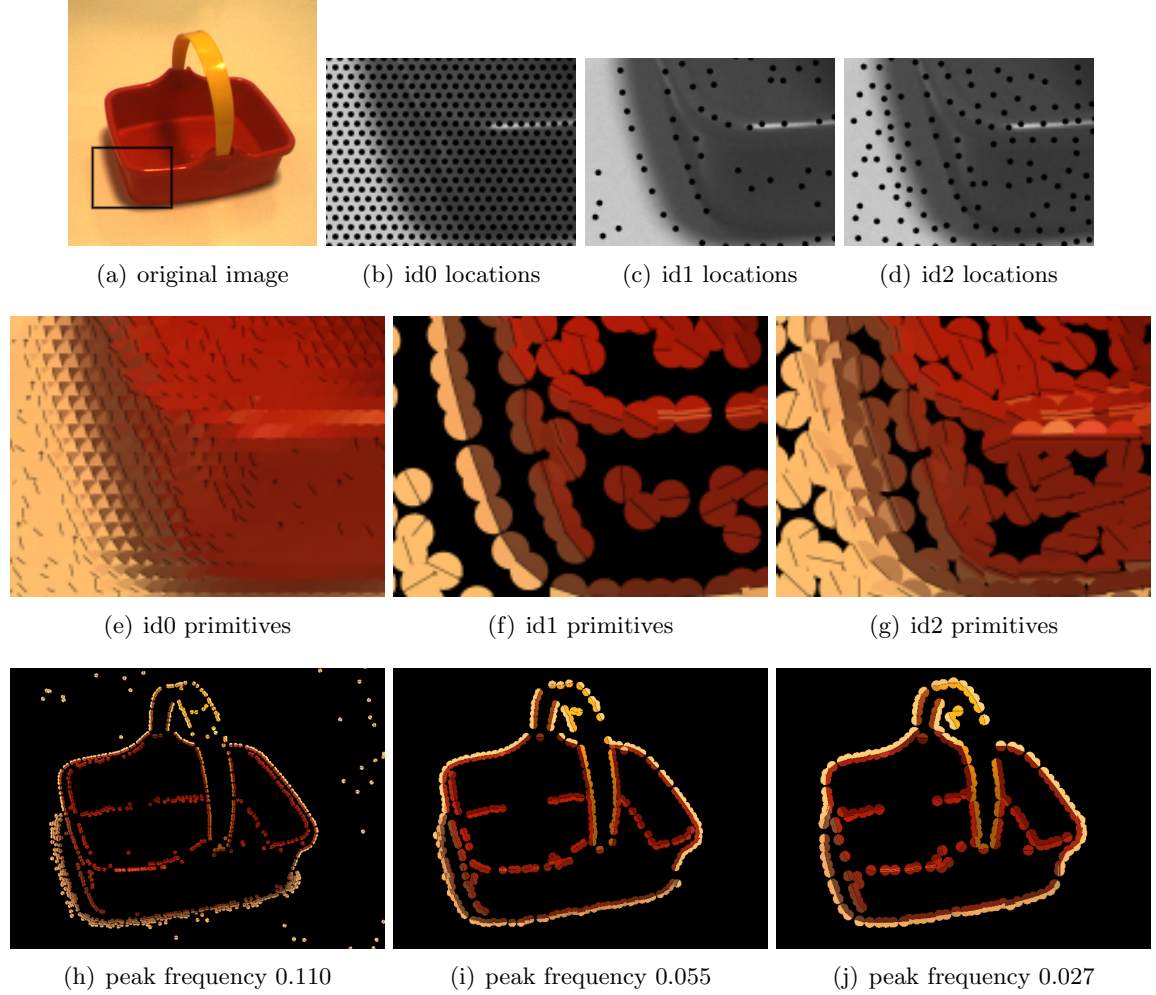


Figure 7: Top row: (a) one image of an object. The black square indicates some detail of the image illustrated in figures (b,c,d,e,f,g); (b,c,d): positions associated to the primitives assuming different intrinsic dimensionality (from left to right, (b) id0, (c) id1 and (d) id2). Middle row (e,f,g): primitives in each of those cases (from left to right, (e) id0, (f) id1 and (g) id2). Bottom row (h,i,j): primitives extracted (from the full image) in contour mode, with origin variance  $> 0.3$  and line variance  $< 0.3$ , for different peak frequencies (from left to right, (h) 0.110, (i) 0.055, and (j) 0.027).

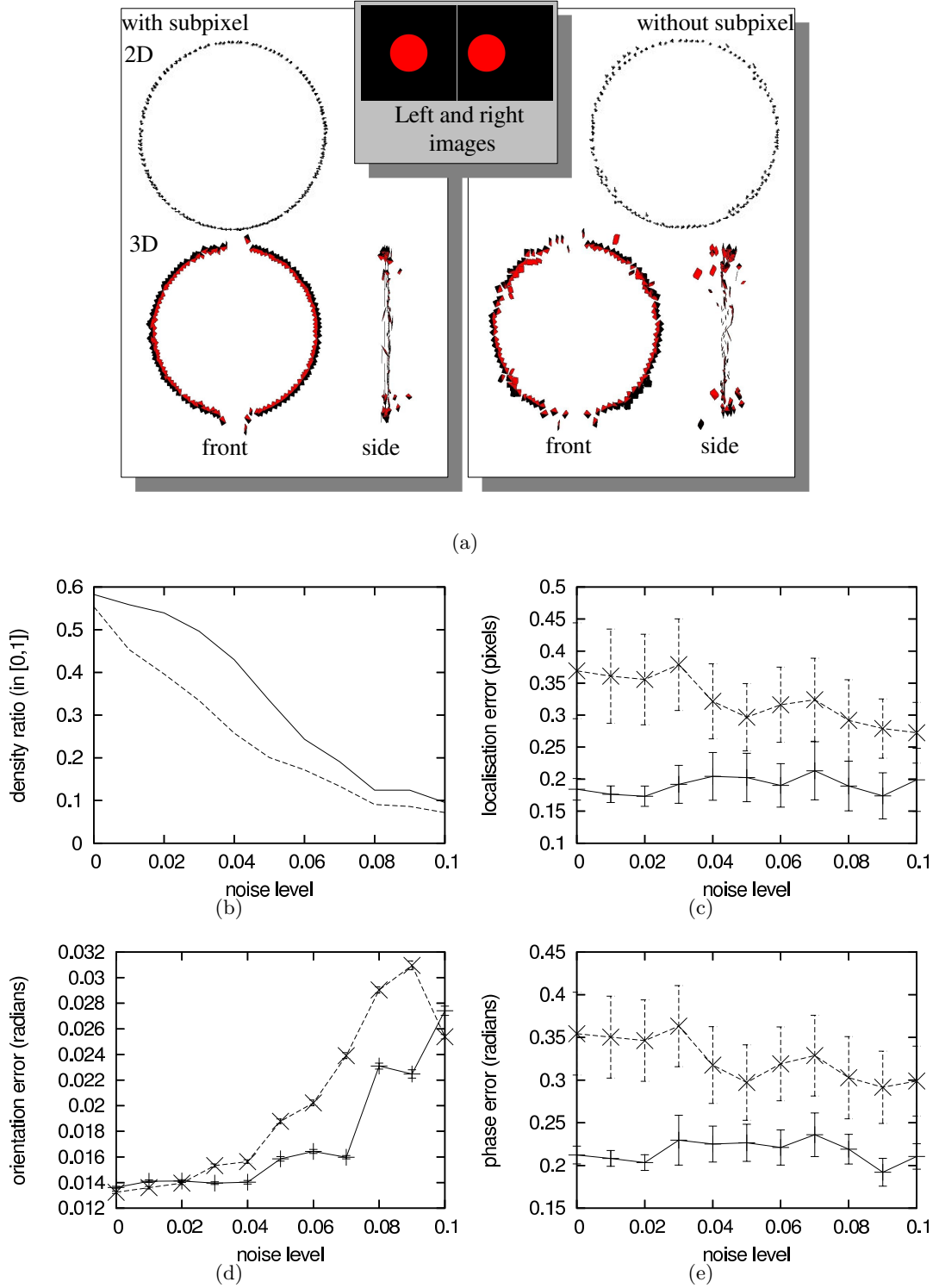


Figure 8: Inset: the pair of synthetic images used for this measurement. (a) 2D- and 3D-primitives extracted from the inset images, respectively with (left) and without (right) sub-pixel localisation. (b,c,d, and e): report the density and accuracy in localisation, orientation and phase of the primitives. The horizontal axis shows the noise level (a noise level of 1 stands for 100% Gaussian noise) added to the image prior to primitive extraction. The solid line shows the accuracy with sub-pixel localisation and the dashed line without. Error bars in (c,d, and e) show the variance.

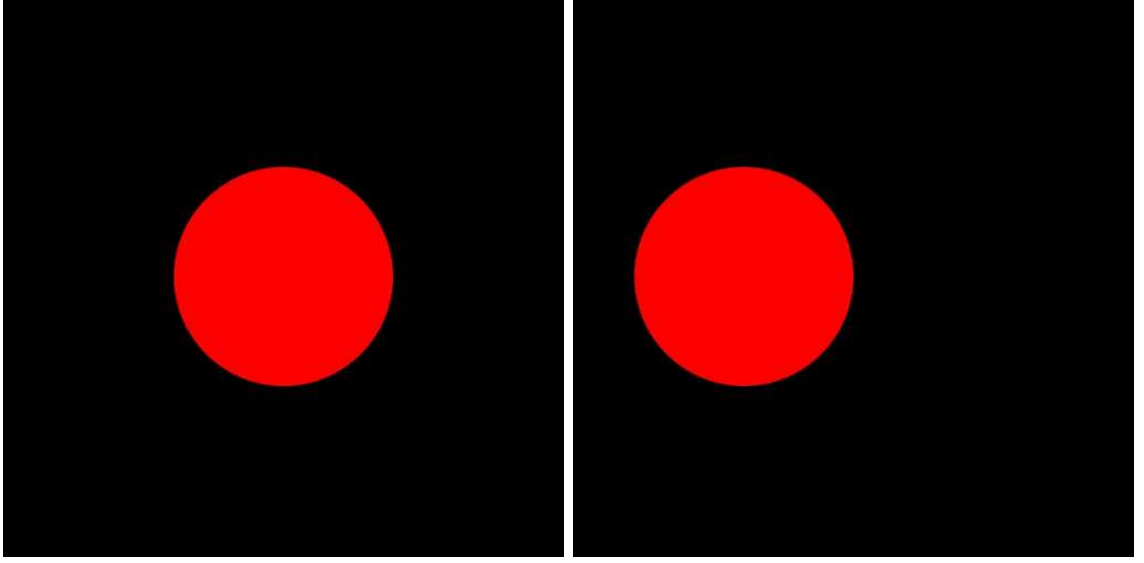


Figure 9: Artificial sequence used to evaluate the accuracy of primitive extraction (see Fig. 8).

should be represented by two edges with phase  $\pm\frac{\pi}{2}$ . Hence, the elimination process should not eliminate these ‘independent’ edges although they can be rather close to each other. The limit of separability is the line/edge bifurcation distance  $d_{leb}$  defined above.

- Distant, yet redundant, putative positions should be discarded. Due to the kernels spatial extent, a given image structure will generate significant response within a radius  $d_k$  that is larger than  $d_{leb}$ . As a consequence, eliminating candidates closer than  $d_{leb}$  preserves all distinct edge structures, plus numerous redundant structures. Conversely, eliminating candidates with a distance smaller than  $d_k$  discards all redundant, plus some distinct structures.

We tackle this problem by a two stage elimination process described in sections 3.2.1 and 3.2.2.

### 3.2.1 Elimination based on the line/edge bifurcation distance $d_{leb}$

First, all candidates  $\mathbf{x}^{(k,l)}$  become ordered according to the associated amplitude  $m(\mathbf{x}^{(k,l)})$ . Starting with the candidates with highest local amplitude, we discard all other candidates  $\mathbf{x}^{(k',l')}$  within a radius  $d_{leb}$ .<sup>6</sup> Since we order the candidates according to the local amplitude, a candidate corresponding to a stronger structure suppresses candidates with weaker structure. Thereby, all non-distinct edges (according to the line/edge bifurcation distance) become deleted but redundant edges are still being preserved. In Fig. 10b, we see that many spurious candidates remain after the first elimination process that are caused by edges with distance smaller than  $d_k$  (see section 3.2.2).

### 3.2.2 Elimination based on the influence radius $d_k$

The local magnitude can be significantly affected by image structures within a radius  $d_k$ . In the second elimination step, starting again from the candidates with the highest local amplitude, the distance between pairs of remaining candidates is compared to  $d_k$ , empirically approximated by  $d_k = 2.2d_{leb}$ . For a pair of intrinsically two-dimensional structures it is sufficient to have a distance

<sup>6</sup>Note that for the quality of the process it is important that all positions are computed with sub-pixel accuracy already at this stage.



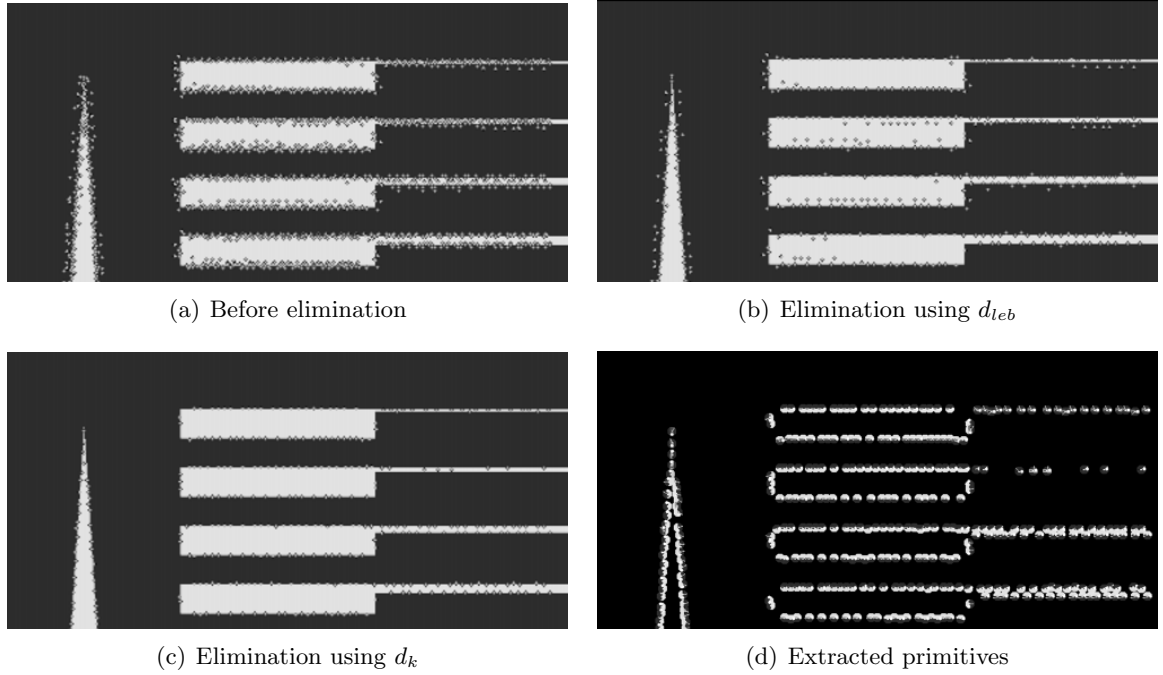


Figure 10: Three stages of the elimination process and the final primitive representation.

smaller than  $d_{leb}$  since they naturally represent maxima in the amplitude representation [27]. For an intrinsically one-dimensional structure, there will be a slant in the local amplitude surface at the redundant structure reaching its maximum at the edge/line structure and decreasing with distance from the edge (see Fig. 5 and Fig. 11). This slant can be checked to distinguish spatially close yet independent structures, that we want to keep, and nearby redundant structures, that we want to discard: For each candidate in a pair with distance smaller  $d_k$ , we test whether the structure is an amplitude maximum, along a line orthogonal to the local orientation (see Fig. 11). This is achieved by comparing each candidate’s amplitude to its direct neighbours, on both sides of the edge, as indicated by the local orientation.<sup>7</sup> Then, redundant structures, *i.e.*, candidates that are not local maximum, are discarded.

The result of this second elimination stage is shown in Fig. 10c, and the resulting primitives in Fig. 10d. Fig. 12 shows the primitives extracted for an artificial test image, for different scales. The image in Fig. 12a shows vertically alternating black/white step-edges, getting narrower towards the right of the image. The primitives extracted at the three scales, for peak frequencies of 0.110, 0.055 and 0.027, are shown in Fig. 12b, c and d, respectively. The effect of the double elimination process at different scales can be seen in this figure. For example if all of the narrower step edges to the right of the image are distinctly extracted in Fig. 12b, only one of the two is extracted in Fig. 12c, while in Fig. 12d the same edges become intrinsically two-dimensional and are not extracted anymore.

### 3.3 Association of visual attributes and confidences

We then associate visual attributes to the remaining positions  $\mathbf{x}^i$ : orientation  $\theta$ , phase  $\varphi$ , and optic flow  $\mathbf{f}$  are computed pixel-wise using filter processes of spatial extent  $d_k$ . Since positions

<sup>7</sup>Note that the criterion ‘local maxima’ that is applicable for id2 structures can not be applied since edge like structures form a ridge in the local amplitude surface (see Fig. 5).

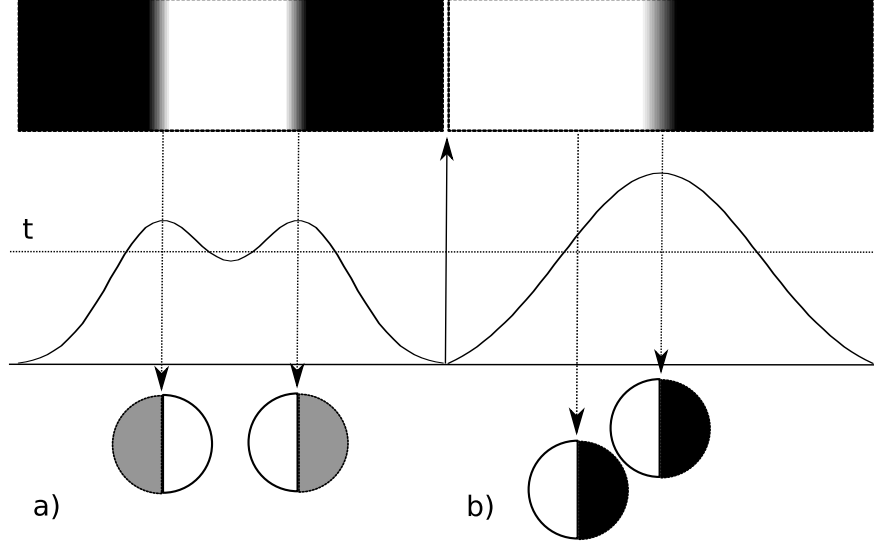


Figure 11: Extraction of redundant primitives due to the slant in the amplitude surface. (a) case of a valid double edge, two primitives are correctly extracted; (b) case of an erroneous extraction of a redundant primitive: because of the mild decay of the amplitude curve, the same structure can cause the extraction of a primitive at a location far from the original structure, (where the amplitude of the response there is still above a given threshold  $t$ ).

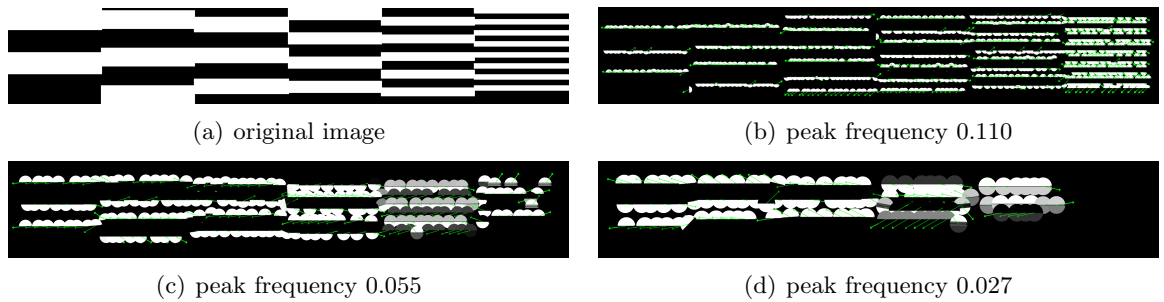


Figure 12: Illustration of the primitives' sampling density: (a) shows an image with gradually (from left to right) narrower white and black bars; (b,c, and d) show the primitives extracted for different peak frequencies.

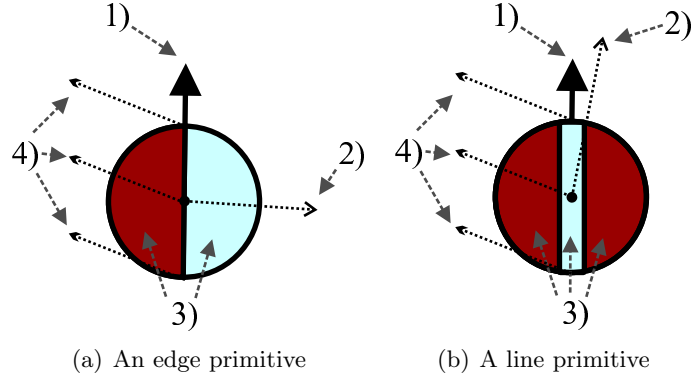


Figure 13: Illustration of the symbolic representation of a primitive for a id1 interpretation, for (a) a bright-to-dark step-edge (phase  $\varphi \neq 0$ ) and (b) a bright line on dark background (phase  $\varphi \neq \frac{\pi}{2}$ . 1) represents the orientation of the primitive, 2) the phase, 3) the colour and 4) the optic flow.

are computed with sub-pixel accuracy, we can also interpolate sub-pixel values for orientation, phase, and optic flow using bi-linear interpolation. Let  $\tilde{x}_0$  and  $\tilde{y}_0$  be the positions computed with sub-pixel accuracy (see section 3.1); let  $\delta_x$  and  $\delta_y$  be the distance to the discrete lower pixels  $x_l$  and  $y_l$ , and  $x_h = x_0 + 1$  and  $y_h = y_0 + 1$ ; then the bi-linear interpolation computation leads to the formula:

$$\tilde{\theta}(\tilde{x}) = \hat{\theta}(x_l, y_l)(1 - \delta_x)(1 - \delta_y) + \hat{\theta}(x_l, y_h)(1 - \delta_x) * \delta_y \quad (8)$$

$$= \hat{\theta}(x_h, y_l)\delta_x(1 - \delta_y) + \hat{\theta}(x_h, y_h)\delta_x\delta_y \quad (9)$$

Note that for the interpolation of orientation and phase, the specific topology of the orientation/phase space needs also to be taken into account. Hence,  $\hat{\theta}$  is transformed such that the distance between all pairs of the set  $\hat{\theta}(x_l, y_l), \hat{\theta}(x_l, y_h), \hat{\theta}(x_h, y_l), \hat{\theta}(x_h, y_h)$  is smaller than  $\frac{\pi}{2}$  and  $\hat{\theta}(\tilde{x})$  is in  $[0, \pi)$ .

For the test picture shown in Fig. 8 we get a localisation error in the area of 0.1 pixel (*i.e.* improvement by a factor four). Bi-linear interpolation of orientation and phase, based on the the sub-pixel positioning, leads also to improvements of a factor 2 and 6, respectively (on the highest frequency level). The effect on reconstruction is also demonstrated in Fig. 8.

Although colour information is available at each pixel position, it is heavily redundant, especially for id0 and id1 signals. For a step-edge structure ( $\frac{\pi}{4} < |\varphi| < \frac{3\pi}{4}$ ) it is natural to distinguish between the colour on each side of the edge ( $c_l, c_r$ ) whilst for a line structure ( $|\varphi| \leq \frac{\pi}{4}$  or  $|\varphi| \geq \frac{3\pi}{4}$ ) the colour of a middle strip  $c_m$  (*i.e.* on the actual line) should also be coded (see Fig. 5c-e and 13). We discussed in section 2.2, the phase can distinguish among these two cases.

Thus we obtain a parametric description of local image patches that we call primitive  $\pi_i$ . For a step-edge this representation is

$$\pi_i = (\mathbf{x}_i, \theta(\mathbf{x}_i), \varphi(\mathbf{x}_i), (c_l(\mathbf{x}_i), c_r(\mathbf{x}_i)), \mathbf{f}(\mathbf{x}_i)) \quad (10)$$

and for line structures

$$\pi_i = (\mathbf{x}_i, \theta(\mathbf{x}_i), \varphi(\mathbf{x}_i), (c_l(\mathbf{x}_i), c_m(\mathbf{x}_i), c_r(\mathbf{x}_i)), \mathbf{f}(\mathbf{x}_i)). \quad (11)$$

The primitives' parameters are explicit and the set of primitives provides a condensed representation of the image. The condensation factor can be computed by the ratio of the number of bits needed

to store the local image patch a primitive stands for. For the highest frequency, such a primitive represents a local image patch of a radius of appr. 3 pixels (*i.e.* if one considers a RGB colour image:  $\pi \cdot 3^2 \cdot 3 \approx 85$  values). The primitive has a dimension of 10 for an edge like structure and 13 for a line-like structure (because optic flow encodes temporal information, it is disregarded). Thus, encoding a primitive, at the highest frequency level, requires maximally 13 bytes, compared to 85 bytes in the original image, leading to a condensation rate of  $d_{co} \approx 85\%$ . Analogously, we get a condensation rate of  $\approx 94\%$  and  $\approx 97\%$  for the other two frequency levels. Note when considering 3D-primitives (see section 4) the condensation rate further increases.

Table 1 shows all parameters included in the primitive extraction. Note that these parameters are either derived from the line/edge bifurcation distance ( $d_{leb}$ ), non-critical ( $w_s$ ), or based on decisions involving a trade off between computational complexity and precision ( $d_k$ ).

## 4 Computation of 3D-primitives

So far we have presented multi-modal image descriptors that code 2D information. However, these descriptors represent visual events occurring at a certain 3D position in space. This depth information is essential for higher level processes because of two reasons. First, humans and robots act in a 3D world where depth information is valuable for, *e.g.*, navigation or grasping. Second, since many structural dependencies of visual events (*e.g.*, rigid body motion) are working on 3D structures, 3D information is essential their formalisation, and for the disambiguation processes they underlie (see [15]).

In the following, we describe an extension of the image primitives to spatial primitives. Thus, the semantic information coded in the image primitives is transferred into the 3D domain.

Given a pair of corresponding points (see [15]) between the left and right images, a meaningful 3D interpretation of this stereo-pair is a 3D point. Contours, however, hold a 2D orientation, and therefore 3D-primitives need to encode the reconstructed 3D orientation  $\Theta$  beside the 3D position  $\mathbf{X}$ . This orientation is computed as the intersection of two planes in space, each defined by the optical centre of one camera and the line in the image plane described by the image primitive's position and orientation — see Fig. 14. The intersection of these two planes in space is a 3D line that provides us with the orientation of the 3D-primitive. In [37], it was shown that using line correspondences for the reconstruction of 3D orientation was generally more accurate than point correspondences.

Phase  $\Phi$  and colour  $\mathbf{C}$  are reconstructed in space as the mean value between the two corresponding image primitives:  $\Phi = \frac{1}{2}(\varphi^L + \varphi^R)$ , and  $\mathbf{C} = \frac{1}{2}(\mathbf{c}^L + \mathbf{c}^R)$

Furthermore, these two modalities encode surface information (respectively contrast and colour transition across an edge); thus, we need to define a 3D surface patch onto which these apply. Unfortunately, it is not possible to reconstruct the exact surface from local information: for a pure id1 signal, the surface on one side does not allow finding the additional correspondence that would be required for the reconstruction of a 3D surface. Moreover, in case of a depth discontinuity, the colour information might come from a 3D position that is completely independent from the 3D orientation information (*i.e.*, the background).

We propose to define as *a priori* 3D surface the plane that is most stable under small viewpoint variations (see Fig. 14). This surface is computed using the 3D orientation of the primitive and an additional *Local Surface Guess* vector  $\Gamma$ , that is defined as follows:

$$\Gamma = \Theta \times V_{pov}, \quad (12)$$

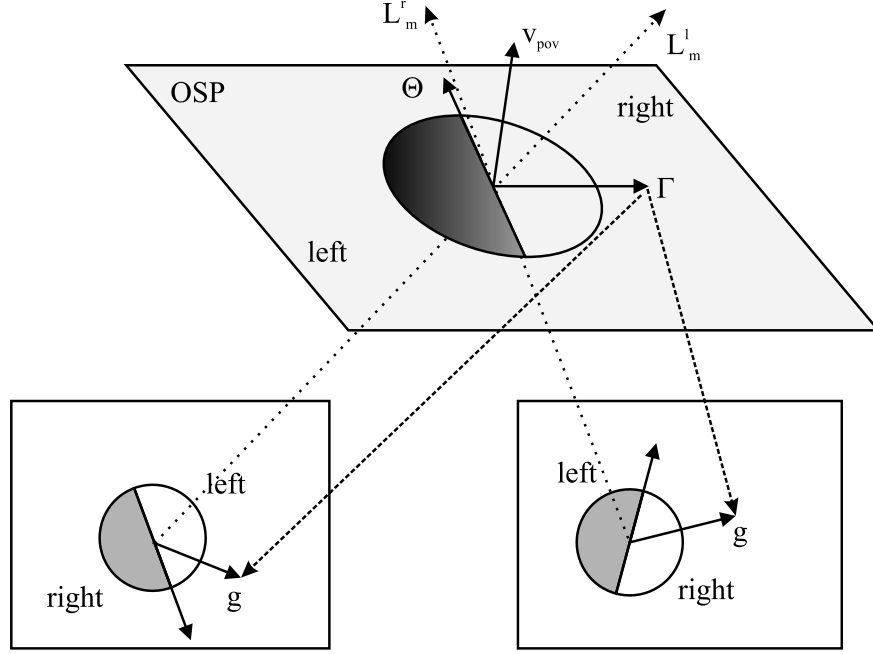


Figure 14: Illustration of the reconstruction of a 3D-primitive from a stereo pair of 2D-primitives.

such that the surface is normal to  $V_{pov}$ , and  $V_{pov}$  is defined as follows:

$$V_{pov} = \frac{1}{2} \left( \overrightarrow{C_L X} + \overrightarrow{C_R X} \right), \quad (13)$$

where  $\overrightarrow{C_L X}$  and  $\overrightarrow{C_R X}$  are the two optical rays joining the location of the primitive  $X$  with the optical centre of the left ( $C_L$ ) and right ( $C_R$ ) camera. The vector  $\Gamma$  also identifies each side of the 3D line, which is critical for modalities like colour and phase that describe the modality transition across the contour.

These allow reconstruction of spatial primitives  $\Pi^{(i,j)}$  each having the parametric description:

$$\Pi^{(i,j)} = (X, \Theta, \Phi, (C_L, C_m, C_r)). \quad (14)$$

The  $j$  index represents the alternative 3D entities generated from different stereo correspondences in the right image to the  $i^{th}$  primitive in the left image. Since a final decision can usually not be made solely based on local information, multiple hypotheses are kept at this stage. In the following section, we will describe different approaches to overcome this ambiguity.

Fig. 8a, bottom, shows front and side views of the 3D primitives reconstructed with (left) and without (right) sub-pixel localisation. The side view offers a better visualisation of depth estimation's quality.<sup>8</sup> It is visible in these images that the sub-pixel localisation of the primitives described in section 3.1 allows for a notably better 3D-reconstruction. The effect of sub-pixel accuracy, for a real scene, is illustrated in Fig. 15, where (a) and (b) show the stereo pair of images that were used, (c) and (d) the 2D-primitives extracted from the left image, with and without sub-pixel accuracy, and (e) and (f) the 3D-primitives reconstructed in both cases.

<sup>8</sup>Note that the accuracy of the depth estimates decreases for horizontal structures. This is due to the ambiguity in reconstructing lines parallel to the epipolar line.

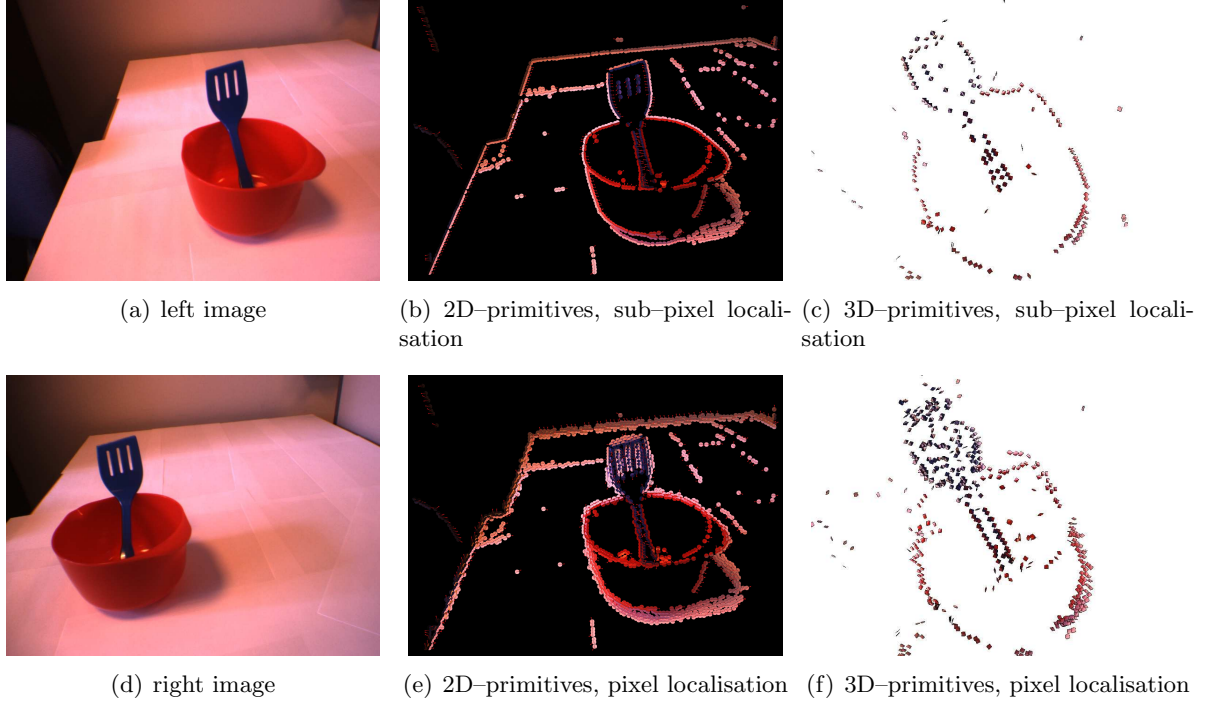


Figure 15: Reconstruction of 3D-primitives in a real scenario. (a) and (d) show the pair of stereo images, (b) (resp. (e)) the 2D-primitives extracted with (resp. without) sub-pixel localisation, and (c) (resp. (f)) the spatial primitives reconstructed with (resp. without) sub-pixel localisation.

## 5 Applications

The primitive representation introduced in this paper has been applied in various contexts (briefly described in this section) and has been part of three different European projects [38–40] in the area of cognitive vision and robotics. The computation of the 3D primitives (this includes computation of 2D primitives in a left and right image (512x512 pixels) as well as the stereo matching and the reconstruction) takes currently between 1 and 2 seconds on a PC<sup>9</sup>.

The primitives described so far are condensed localised descriptors with explicit semantics, and therefore, symbolic descriptors of a local scene structure. Since the primitives are processed locally, they are necessarily as ambiguous as the locally computed modalities that they code. However, a number of relations defined upon the primitives (described in the next sub-section) can be used to disambiguate the local information using the global context.

### 5.1 Relations and operations defined on primitives

Since primitives are a symbolic description of local image patches, the relations and operations defined on a primitive provides the context wherein information is processed. Here, we briefly provide four definitions of primitives' second order relations: collinearity, rigid body motion, coplanarity and co-colourity (see also Fig. 16a).

<sup>9</sup>Computational time depends for example on the amount of primitives which depend on the image structure

### 5.1.1 Good continuation (collinearity)

In [15], a measure of the likelihood for two 2D-primitives to belong to the same image contour  $C(\pi_i, \pi_j)$  is defined (see Fig. 16a,i). This allows for the definition of a stereo constraint (see, *e.g.*, [41, 42]) that makes use of local image information (as encoded by the primitives) as well as contextual information gathered from other primitives in the vicinity (see [15]). The collinearity constraint can naturally be extended to 3D-primitives ( $C(\Pi_i, \Pi_j)$ ) by applying the following rule:

$$C(\Pi_{i,p}, \Pi_{j,q}) \equiv C(\pi_i, \pi_j) \wedge C(\pi_p, \pi_q), \quad (15)$$

where  $\Pi_{i,p}$  and  $\Pi_{j,q}$  are the 3D-primitives reconstructed from the stereo pairs  $(\pi_i, \pi_p)$  and  $(\pi_j, \pi_q)$ , respectively.

### 5.1.2 Rigid body motion

The change of position and orientation induced by a rigid body motion ( $\text{RBM}(\Pi)$ ) can be computed analytically (see, *e.g.*, [43]); phase and colour can be approximated to be constant (see Fig. 16a,iv). In [?] we used a simple scheme to track primitives over time (using the optic flow information) and used it to estimate the camera motion from our visual representation, assuming the absence of independently moving objects.

### 5.1.3 Co-planarity

The relation co-planarity  $\text{Cop}(\Pi_i, \Pi_j)$  between two 3D-primitives (see Fig. 16a,ii) indicates the likelihood of these two primitives to be part of the same surface (see section 5.4) and suggests a way to grasp the object the primitives' pair belongs to (see section 5.3).

### 5.1.4 Co-colourity

The relation co-colourity (see Fig. 16a,iii) expresses the similarity between two primitives' colour.<sup>10</sup>

Semantic relations are used at a stage of processing after the condensation step (called early cognitive vision in [6]), in the following manners:

- predictions between visual events become formulated (such as the change of a local image patch under motion or the likelihood of being part of the same collinear group) and by that the locally ambiguous information becomes, disambiguated (see [15]),
- sets of primitives can become connected to higher visual entities such as 3D surfaces (section 5.4) and objects (section 5.2),
- low-order combinations of primitives become associated to robot actions such as grasping (section 5.3).

---

<sup>10</sup>For each primitive only the colour component on the inner side of the surface defined by the pair of primitives is considered.

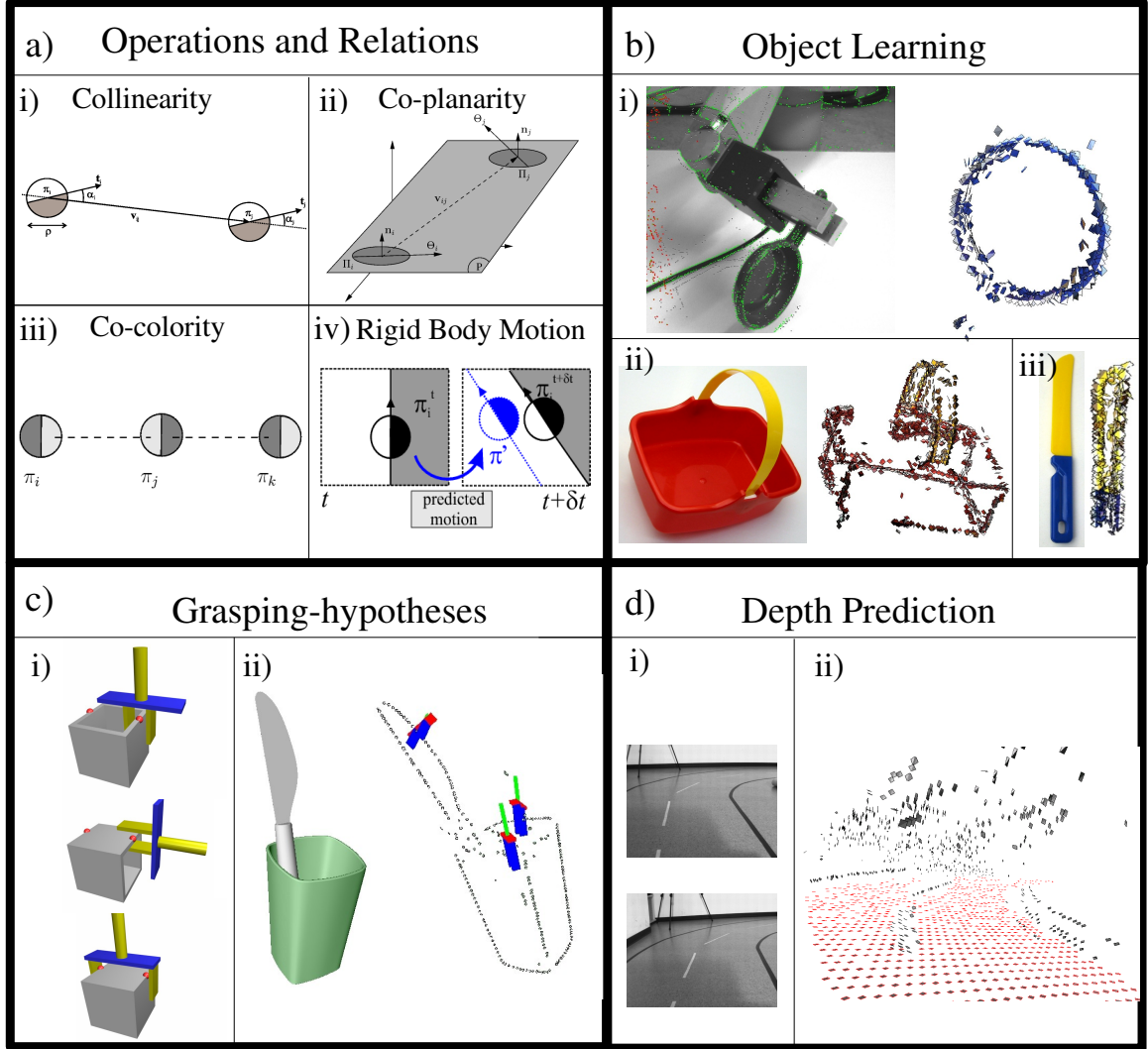


Figure 16: (a) Relations defined on the multi-modal primitives (b) Extraction of object representations. (c) Grasping options generated by second order relations of primitives. i) Three of the elementary grasps that can be inferred from one pair of co-planar primitives (identified by the two red dots on the object). ii) left: One synthetic scene; right: the 3D-primitives reconstructed and three examples of the grasps inferred by the system described in [44]. (d) Depth predictions based on co-planarity relations (note since in the stereo images occur rather large disparities there is a certain amount of outliers which however do not effect the surface prediction).



## 5.2 Object model learning

Our visual representation was used to learn object shapes. The object is manipulated by a robotic arm in front of a pair of stereo cameras. Since the motion of the robot arm is known, and the stereo and robot system are properly calibrated, we can use the RBM relation described above to track 3D-primitives describing the object held by the arm in a robust manner. Thus, we can infer that 3D-primitives that do not move according to the motion of the arm are therefore not part of the manipulated object. Furthermore, object features that were not initially observed can be added to the object representation. Therefore, this algorithm allows us to:

- remove spurious 3D-primitives from the object model, and
- complete the object model using information from all available viewpoints.

Assuming that the arm’s motion spans adequately the object’s pose space, a full 3D model of the object can be generated by this procedure [16].

This is illustrated in Fig. 16b), where i) shows the robotic setup holding a pan-like object. The green dots show the 3D-primitives that were successfully tracked over time, whilst red and black dots show the primitives that were not. On the right hand side, the learnt object model is shown, from a different viewpoint.<sup>11</sup> Then panels ii) and iii) show the shape model obtained for two different objects.

## 5.3 Generating grasping hypotheses

Our representation has also been used to define grasping options in a scene (see Fig. 16c) and [44]). Essentially, co-planar primitives (supported by the relations collinearity and co-colourity) define planes that are good candidates for an initial grasping hypothesis. Fig. 16c,i) shows three examples of grasping hypotheses generated from a single pair of co-planar 3D-primitives. Fig. 16c,ii) shows, on the left, one image from a scenario created using the grasping simulation software GraspIt, that was also used for the evaluation of our approach (for details, see [44]). On the right, we see the 3D-primitives reconstructed from this scene, alongside three of the candidate grasps generated by our system on this scenario (shown from a different viewpoint than the image).

If evaluated as successful by haptic information, such a grasping action gives the physical control over objects required for the object learning sketched in section 5.2. This provides a robot with a basic exploratory behaviour: 1) try to grasp at the (unknown) environment; 2) if successful, manipulate the object; 3) learn a full 3D representation of the object.

Such a behaviour enables a naive robot to progressively learn an internal representation of the world with only minimal prior world knowledge. This is relevant in the context of the European project PACO+ [38].

## 5.4 Depth prediction at homogeneous image areas

The primitives introduced here represent id1 structures. It is known that it becomes increasingly difficult to find correspondences between local patches the more they lack structure (*i.e.* tending toward the id0 corner of the iD triangle, see Fig. 2.1). On the other hand, it is known that lack of structure also indicates lack of a depth discontinuity [25,45]. Moreover, it was statistically shown in [46] that co-planarity allows predicting depth at homogeneous image surfaces (see Fig. 16d). Such a scheme can be used to ‘fill in’ our representation at homogeneous surfaces using co-planar

---

<sup>11</sup>The gap in the representation, on the handle of the pan, is a part of the object occluded by the gripper. The model could be completed by using at least one alternative grasp.

relationship between id1 primitives: in Fig. 16d, the homogeneous primitives inferred using such a scheme are shown with a red border on each predicted homogeneous primitive. One can see that the whole road becomes inferred from the reconstructed lane markings. Note also that spurious 3D-primitives (reconstructed due to nearly horizontal structures in the images) do not generate hypotheses due to their inherently random distribution.

## 6 Discussion

In this section, we give some information about the context of our representations in terms of their role in a cognitive vision architecture and the biological analogy of a primitive to a hypercolumn ([3]). Furthermore, we discuss the relation of our feature descriptor to other visual descriptors and give some indications about current and future work.

### 6.1 Primitives as part of a cognitive vision system

The primitives introduced in this paper are one pillar in the Early Cognitive Vision paradigm described in, *e.g.*, [47], developed in the context of the European project ECOVision [39]. It is now applied within two other European projects addressing higher level tasks such as scene interpretation in a driving assistance scenarios (Drivscio [40]) and cognitive robotics (PACO+ [38]). While in the ECOVision project, the primitives were used for the disambiguation of local information and outlier removal using contextual knowledge (see, *e.g.*, [48]), in Drivscio we address general 3D scene interpretation tasks such as the explicit structuring of visual information in terms of larger entities and the linking of such entities to driving actions. We also address classical computer vision tasks such as object model learning, pose estimation, and object recognition. In addition to these tasks, in PACO+ we interface our representations with a robot's actions [16] and with a high level planner [49]. The broad applicability of this representation stems from the ECOVision project's goal to develop a general vision machine, in analogy to the human visual system (see subsection 6.2). In particular, we were interested in allowing for a semantic interpretation of visual scenes. For this, we believe that a transition of the representation of visual information to a symbolic level is required and that this transition is driven by the two properties *Predictability* and *Condensation* mentioned in the introduction. This allows us to formulate strong and efficient predictions coded in the relations described in subsection 5.1 that can be used to disambiguate the information as well as to bridge to higher level representations of objects and relations of objects to actions.

The transformation of visual information to a symbolic level as done in the condensation process described in section 3 can be motivated by three drawbacks of *pixel-wise* interpretation of visual sub-aspects such as orientation, phase, and optic flow (a more detailed discussion is given in [50]).

#### 6.1.1 Low predictability

Disambiguation requires predictions of events as a consequence of other events. In [14], we showed that predictability on the pixel level is weaker than on a level where attributes with richer semantic content are computed. For example, certain *Gestalt* laws such as collinearity and parallelism can only be found in visual data when making use of orientation instead of the actual pixel value [11–13]. Going beyond, we showed in [51] that the statistical dependencies of local line segments corresponding to these *Gestalt* laws become much more pronounced within our multi-modal representation.

### 6.1.2 Ill-defined semantic of features associated to position

The computation of a feature descriptor is based on information that covers a larger spatial area. For example, to estimate orientation we need at least three samples. In general, depending on their bandwidth, filters cover large spatial areas to achieve higher precision of estimates. Therefore local orientation's estimation from linear filters suffers from the superposition of the true orientation with values from other structures in the vicinity, that become more prominent with the distance. Hence, the orientation computed close to an edge still depends on this edge but also on other surrounding structures and therefore the orientation at a pixel position that is not located precisely on an edge is ill-defined. As a consequence, beside the need to condense the image information for establishing contextual processes, the features themselves need to be associated to discrete locations in the image. Hence, in our representation, the primitives' position is defined by a dynamic process, and visual attributes are associated to this position.

### 6.1.3 Cross-connection of modality processing

Visual modalities' processing can be supported by cross-connections between modalities. For example, a reasonable colour coding depends on the orientation and phase. Since a step edge distinguishes between two areas of different colour it makes sense to code these two values separately. On the other hand, for a line-like structure there are three areas we need to distinguish (see Fig. 13). Therefore, we need to understand the feature extraction process as a recurrent process wherein the computation of individual modalities interact with each other. By using the primitives, we make use of such cross-connections for example for colour and depth processing (see section 5.4).

## 6.2 Biological analogy

Primitives have a direct biological analogy, discussed in detail in [52], that we will summarise here. The main information stream in the human visual system projects to area V1 in the cortex [53]. The structure of V1 has been investigated by Hubel and Wiesel in their ground-breaking work [1,3]. V1 is organised as a retinotopic map that has a specific repetitive pattern of substructures called hyper-columns. Hyper-columns themselves contain so called orientation columns and blobs which are mainly involved in colour processing. However, in an orientation column, we find cells sensitive, beside orientation, to disparity [5,54], local motion [55], colour [3], and phase [56]. Also specific responses to junction-like structures could be measured [4]. Therefore, it is believed that V1 processes local feature descriptions, analogous to the primitives which can be regarded as functional abstractions of hypercolumns. Moreover, there is a high (feed-forward and feedback) connectivity, within V1 and towards other visual areas. This is thought to be the basis for the processing of contextual information [53]. Such connectivity is analogous to the contextual information gathered from the primitives' relations defined in 5.1.

## 6.3 Relation to other local descriptors

Feature extraction from images is the combination of two distinct, yet dependent, processes (see, e.g., [57,58]): first comes the detection of *interest points*, which are locations in the image likely to contain information (this is required to obtain a sparse feature map); second, the information encoding at these locations into *feature descriptors*. There has been a large amount of work on both of these aspects.

A prominent example of an interest point detector is the Harris corner detector [59], and the scale adapted *Harris-Laplace* detector proposed by [58], which extract features at locations that

have maximal local variations in space. This can be compared to the concept of intrinsically two-dimensional points presented in section 2.1. Edge detectors, like Canny’s classical algorithm [60], zero-crossings [7], or phase congruence [30], return edge pixels (similar to intrinsically one-dimensional points in section 2.1). The *Hessian-Laplace*, localise points in space at the local maxima of the Hessian determinant and in scale at the local maxima of the Laplacian of Gaussian. This detects *blob-like* structures, that could be compared to intrinsically zero-dimensional structures presented in section 2.1.

The extracted features should be robust (ideally invariant) under illumination and viewpoint changes, while remaining distinct. In other words, an ideal feature descriptor allows for a metric such that: features extracted from the same 3D area under different perspectives are proximate, and features originating from different 3D areas are distant. Although it has been shown that such a metric, in the general case, do not exist [61], the recent years have seen the development of robust and affine invariant descriptors. In particular, it has been shown that SIFT features [62], and derivatives such as GLOH [58], are very efficient for a large set of matching tasks including: multiple view reconstruction [63], object recognition [62], pose estimation [64], and image retrieval [65].

However, although we recognise the importance of invariance in computer vision, this is not the primary motivation for our representation, but rather our goal is to initiate a process wherein scene structures’ geometric and appearance information become represented *explicitly* in terms of local symbolic descriptors and by semantic relations between them, both in 2D and 3D. Thus we intent to bridge the gap between early image processing and higher stages of visual and cognitive processing that require an abstract symbolic description of the world, as addressed, *e.g.*, in the EU-projects Drivscio [40] and PACOplus [38]. Because our representation’s explicitness, we are able to use the necessary structural knowledge for object’s and action’s representation. In this context, our scene representation based on multi-modal primitives addresses a number of issues in an original way:

### 6.3.1 Multi-modality

primitives cover the main visual modalities established in computer- and human vision and, hence, carry a rich semantic interpretation expressed in local symbols and their relations.

### 6.3.2 Condensation

primitives reduce the dimensionality of image data while preserving its significant aspects (*e.g.*, in [66] we showed that primitives allowed for matching performance comparable to normalised cross-correlation).

### 6.3.3 Different experts for different structures

the interpretation of the local signal by primitives is not static but depends on the intrinsic signal structure, leading to a system of different experts for different signal structures, such as edges, lines, homogeneous patches and corners (as in the human system).

### 6.3.4 Primitives initiate disambiguation

primitives are not a final statement about a scene’s local structure; indeed the *confidence* associated to each primitive as well as its parameters can become modified in disambiguation processes formalising contextual information.

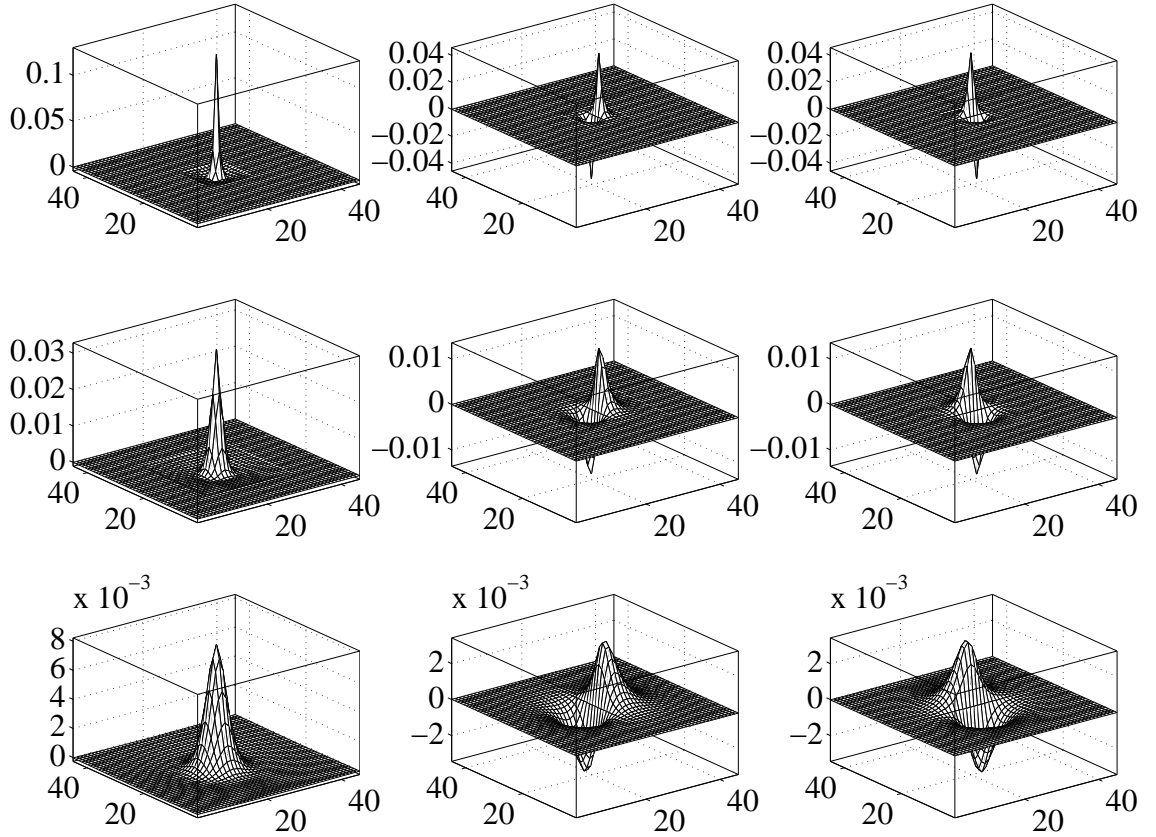


Figure 17: Impulse responses of the DOP filter and its Riesz transforms. From left to right: DOP filter, first Riesz transform, second Riesz transform. From top to bottom: scales (1,2), (2,4), (4,8).

## 6.4 Current and future work

Currently, our system treats different scales independently; this is appropriate since so far we only deal with edge-like structures, that show stable properties across scales. Nevertheless, selecting the optimal scale of processing would reduce memory and computational requirements while improving the overall robustness of the edge representation. An extension of our approach into scale-space where scale itself expressed by a feature (see, *e.g.*, [32, 36, 67]) is being considered.

Furthermore, we intend to introduce symbolic descriptors for different (other than edge) image structures. For homogeneous image patches this has been already discussed in section 5.4. In [26], we have discussed an extension of our approach to junction-like structures. We note that this requires not only a junction detection and interpretation algorithm but also the definition of appropriate relations between different junctions as well as between edges and junctions. We are also currently doing the first steps towards the representation of texture which in particular requires a representation of different scales.

## A Split of identity

Quadrature filters based on the monogenic signal [27] are rotation invariant, *i.e.* they commute with the rotation operator. Hence, for an appropriate choice of polar coordinates, two coordinates do not change under rotations (amplitude and phase), whereas the third coordinate directly reflects the rotation angle. This kind of quadrature filter, which is called *spherical quadrature filter* [19], is formed by triplet of filters: a radial bandpass filter and its two Riesz transforms [21]. As in [19] we construct the bandpass filter from *difference of Poisson* (DOP) filters, in order to get analytic formulations of all filter components in the spatial domain and in the frequency domain. The DOP filter is an even filter (w.r.t. point reflections in the origin) and its impulse response (convolution kernel) and frequency response (Fourier transform of the kernel) are respectively given by:

$$h_e(\mathbf{x}) = \frac{s_1}{2\pi(|\mathbf{x}|^2 + s_1^2)^{\frac{3}{2}}} - \frac{s_2}{2\pi(|\mathbf{x}|^2 + s_2^2)^{\frac{3}{2}}} \quad (16)$$

$$H_e(\mathbf{u}) = \exp(-2\pi|\mathbf{u}|s_1) - \exp(-2\pi|\mathbf{u}|s_2) . \quad (17)$$

For convenience, we combine the two Riesz transforms of the DOP filter in a complex, odd filter, yielding the impulse response and the frequency response:

$$h_o(\mathbf{x}) = \frac{\mathbf{x}_1 + i\mathbf{x}_2}{2\pi(|\mathbf{x}|^2 + s_1^2)^{\frac{3}{2}}} - \frac{\mathbf{x}_1 + i\mathbf{x}_2}{2\pi(|\mathbf{x}|^2 + s_2^2)^{\frac{3}{2}}} \quad (18)$$

$$H_o(\mathbf{u}) = \frac{u_2 - iu_1}{|\mathbf{u}|} (\exp(-2\pi|\mathbf{u}|s_1) - \exp(-2\pi|\mathbf{u}|s_2)) , \quad (19)$$

respectively. The impulse responses of the filters for  $(s_1, s_2) = (1, 2), (2, 4), (4, 8)$  are shown in Fig. 17.

The split of identity (*i.e.* the separation of the signal into local amplitude, orientation and phase) is obtained by switching to appropriate polar coordinates. In particular, we transform the filter responses according to

$$m(\mathbf{x}) = \sqrt{I_e(\mathbf{x})^2 + |I_o(\mathbf{x})|^2} \quad (20)$$

$$\theta(\mathbf{x}) = \arg I_o(\mathbf{x}) \pmod{\pi} \quad (21)$$

$$\varphi(\mathbf{x}) = \text{sign}(\Im\{I_o(\mathbf{x})\}) \arg(I_e(\mathbf{x}) + i|I_o(\mathbf{x})|) , \quad (22)$$

which gives the desired amplitude, orientation, and phase information.

Fig. 18 shows a radial cut through the DOP bandpass filters for a certain range of scales and their superposition, demonstrating a homogeneous covering of the frequency domain. For infinitely many bandpass filters, the superposition is one everywhere, except at the origin. In our system, we apply filters on three frequency levels (see Fig. 17). The applied bandpasses are indicated by the darker colour in Fig. 18.

The local orientation associated to the image patch is described by  $\theta()$ . The orientation parameter  $\theta$  and the phase parameter  $\varphi$  can take values in  $[-\pi, \pi)$  (see figure 4). However, this would lead to a redundant representation since, e.g., a horizontal dark/bright edge can be interpreted as an edge with orientation  $\pi/2$  and phase  $\pi/2$  but also as a bright/dark edge with orientation  $3/2\pi$  and phase  $-\pi/2$ . A parametrisation of orientation between  $[0, 2\pi)$  is usually referred to as direction. However, direction can not be unambiguously estimated locally (see [29, 31]). Therefore, we restrict the orientation values to  $[0, \pi)$ . Another problem, that becomes apparent in Fig. 4 is the singularity in orientation for phase  $\varphi = 0$  and  $\varphi = -\pi$ . Indeed, all orientations are valid close to those singularities. The deeper reason for that is (see IEEE, overcome by averaging).

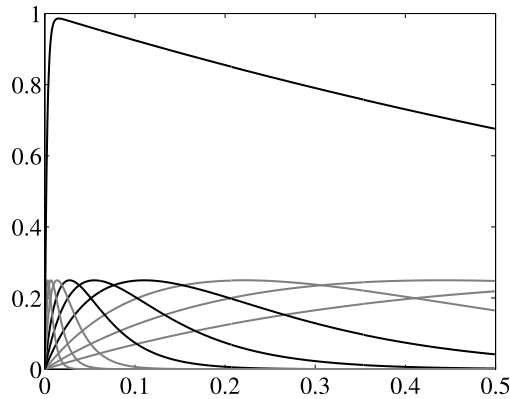


Figure 18: DOP bandpass filters and their superposition approaching the identity (x-axis representing the frequency). The superposition and the filters applied in this paper are indicated by the darker lines.

## Acknowledgement

The work on the multi-modal primitives started in 1998 in Kiel, Germany. It became an important part of the European project ECOVISION (2001–2003) [39] and is now applied in the context of vision based robotics as well as driver assistant systems in the two European projects PACO+ (2006–2010) [38] and Drivscio (2006–2009) [40]. Many master and PhD students have been involved in this project and we would like to thank Markus Ackermann, Emre Baseski, Kord Ehmcke, Michael Felsberg, Christian Gebken, Oliver Granert, Danial Grest, Marco Hahn, Thomas Jäger, Sinan Kalkan, Dirk Kraft, Florian Pilz, Martin Pörksen, Torge Rabsch, Bodo Rosenhahn, Morten Skov, Shi Yan, Daniel Wendorff and Jan Woetzel. We would like to thank in particular and for their contributions to this work.

## References

- [1] D. Hubel and T. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *J. Physiology*, vol. 160, pp. 106–154, 1962.
- [2] M. Oram and D. Perrett, “Modeling visual recognition from neurobiological constraints,” *Neural Networks*, vol. 7, pp. 945–972, 1994.
- [3] D. Hubel and T. Wiesel, “Anatomical demonstration of columns in the monkey striate cortex,” *Nature*, vol. 221, pp. 747–750, 1969.
- [4] I. Shevelev, N. Lazareva, A. Tikhomirov, and G. Sharev, “Sensitivity to cross-like figures in the cat striate neurons,” *Neuroscience*, vol. 61, pp. 965–973, 1995.
- [5] H. Barlow, C. Blakemore, and J. Pettigrew, “The neural mechanisms of binocular depth discrimination,” *Journal of Physiology (London)*, vol. 193, pp. 327–342, 1967.
- [6] N. Krüger, M. V. Hulle, and F. Wörgötter, “Ecovision: Challenges in early-cognitive vision,” *International Journal of Computer Vision*, accepted.
- [7] D. Marr, *Vision*. Freeman, 1982.
- [8] B. Schiele and J. Crowley, “Probabilistic object recognition using multidimensional receptive field histograms,” *Advances in Neural Information Processing Systems*, vol. 8, pp. 865–871, 1996.

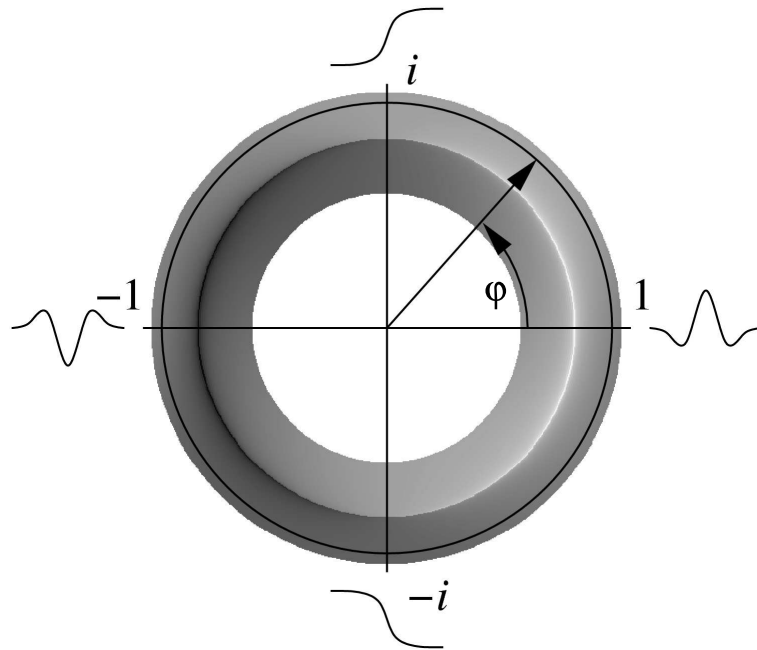


Figure 19: **Left:** Variation of contrast transition according to phase variation. Note that a phase of  $\pi$  codes a dark line on bright background, a phase of  $-\pi/2$  codes a bright/dark edge, a phase of 0 codes a bright line on a dark background while a phase of  $\pi/2$  codes a dark/bright edge. As can be seen the continuum between these cases is also coded by the phase. **Right:** Luminance profiles corresponding to left image.

- [9] M. Lades, J. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Würtz, and W. Konen, "Distortion invariant object recognition in the dynamik link architecture," *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, 1993.
- [10] N. Krüger and F. Wörgötter, "Statistical and deterministic regularities: Utilisation of motion and grouping in biological and artificial visual systems," *Advances in Imaging and Electron Physics*, vol. 131, pp. 82–147, 2004.
- [11] J. Elder and R. Goldberg, "Ecological statistics of Gestalt laws for the perceptual organization of contours," *Journal of Vision*, vol. 2, no. 4, pp. 324–353, 8 2002.
- [12] W. Geisler, J. Perry, B. Super, and D. Gallogly, "Edge Co-occurrence in Natural Images Predicts Contour Grouping Performance," *Vision Research*, vol. 41, pp. 711–724, 2001.
- [13] N. Krüger, "Collinearity and parallelism are statistically significant second order relations of complex cell responses," *Neural Processing Letters*, vol. 8, no. 2, pp. 117–129, 1998.
- [14] P. König and N. Krüger, "Perspectives: Symbols as self-emergent entities in an optimization process of feature extraction and predictions," *Biological Cybernetics*, vol. 94, no. 4, pp. 325–334, 2006.
- [15] N. Pugeault, F. Wörgötter, and N. Krüger, "Multi-modal scene reconstruction using perceptual grouping constraints," in *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06)*, 2006.
- [16] N. Pugeault, E. Baseski, D. Kraft, F. Wörgötter, and N. Krüger, "Extraction of multi-modal object representations in a robot vision system," in *Robot Vision Workshop at the Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, 2007, pp. 126–135.



- [17] H.-H. Nagel, "On the estimation of optic flow: Relations between different approaches and some new results." *Artificial Intelligence*, vol. 33, pp. 299–324, 1987.
- [18] M. Felsberg, S. Kalkan, and N. Krüger, "Continuous characterization of image structures of different dimensionality," *IEEE Transactions on Image Processing (submitted)*, 2006, submitted.
- [19] M. Felsberg, "Low-level image processing with the structure multivector," Ph.D. dissertation, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel, 2002.
- [20] C. Zetzsche and E. Barth, "Fundamental limits of linear filters in the visual processing of two dimensional signals," *Vision Research*, vol. 30, 1990.
- [21] B. Jähne, *Digital Image Processing – Concepts, Algorithms, and Scientific Applications*. Springer, 1997.
- [22] M. Felsberg and N. Krüger, "A probabilistic definition of intrinsic dimensionality for images," *Pattern Recognition, 24th DAGM Symposium*, 2003.
- [23] N. Krüger and M. Felsberg, "A continuous formulation of intrinsic dimension," *Proceedings of the British Machine Vision Conference*, pp. 261–270, 2003.
- [24] S. Kalkan, D. Calow, F. Wörgötter, M. Lappe, and N. Krüger, "Local image structures and optic flow estimation," *Network: Computation in Neural Systems*, vol. 16, no. 4, pp. 341–356, 2005.
- [25] S. Kalkan, F. Wörgötter, and N. Krüger, "Statistical analysis of local 3d structure in 2d images," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1114–1121, 2006.
- [26] S. Kalkan, S. Yan, F. Pilz, and N. Krüger, "Improving junction detection by semantic interpretation," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.
- [27] M. Felsberg and G. Sommer, "The monogenic signal," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3136–3144, December 2001.
- [28] S. Sabatini, G. Gastaldi, F. Solari, K. Pauwels, M. van Hulle, J. Diaz, E. Ross, N. Pugeault, and N. Krüger, "Compact (and accurate) early vision processing in the harmonic space," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.
- [29] G. H. Granlund and H. Knutsson, *Signal Processing for Computer Vision*. Kluwer Academic Publishers, Dordrecht, 1995.
- [30] P. Kovess, "Image features from phase congruency," *Videre: Journal of Computer Vision Research*, vol. 1, no. 3, pp. 1–26, 1999.
- [31] N. Krüger and M. Felsberg, "An explicit and compact coding of geometric and structural information applied to stereo matching," *Pattern Recognition Letters*, vol. 25, no. 8, pp. 849–863, 2004.
- [32] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117–156, 1998.
- [33] H.-H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 565–593, 1986.
- [34] L. Middleton and J. Sivaswamy, *Hexagonal Image Processing : A Practical Approach*. Springer Verlag, 2005.
- [35] R. Staunton and N. Storey, "A comparison between square and hexagonal sampling methods for pipeline image processing," *Proc. SPIE*, vol. 1194, pp. 142–151,, 1989.
- [36] J. H. Elder and S. W. Zucker, "Local scale control for edge detection and blur estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 699–716, jul 1998.
- [37] L. Wolff, "Accurate measurements of orientation from stereo using line correspondence," in *Proceedings of the IEEE Computer Vision and Pattern Recognition conference*, 1989.

- [38] PACO-PLUS, “PACO-PLUS: perception, action and cognition through learning of object-action complexes,” Integrated Project (IST-FP6-IP-027657), 2006. [Online]. Available: <http://www.paco-plus.org>
- [39] ECOVISION, “Artificial visual systems based on early-cognitive cortical processing (EU-Project),” <http://www.pspc.dibe.unige.it/ecovision/project.html>, 2003.
- [40] DrivSco, “DrivSco: learning to emulate perception-action cycles in a driving school scenario,” FP6-IST-FET, contract 016276-2, 2006. [Online]. Available: <http://www.pspc.dibe.unige.it/drivsc/>
- [41] R. Chung and R. Nevatia, “Use of monocular groupings and occlusion analysis in a hierarchical stereo system,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 50–56.
- [42] S. Sarkar and K. Boyer, *Computing Perceptual Organization in Computer Vision*. World Scientific, 1994.
- [43] O. Faugeras, *Three-Dimensional Computer Vision*. MIT Press, 1993.
- [44] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger, “Model-independent grasping initializing object-model learning in a cognitive architecture,” *IEEE International Conference on Robotics and Automation, ICRA07, Workshop: From features to actions - Unifying perspectives in computational and robot vision*, 2007.
- [45] W. Grimson, “Surface consistency constraints in vision,” *CVGIP*, vol. 24, no. 1, pp. 28–51, 1983.
- [46] S. Kalkan, F. Wörgötter, and N. Krüger, “Depth prediction at homogeneous image structures,” Robotics Group, Maersk Institute, University of Southern Denmark, Tech. Rep. 2, 2007.
- [47] F. Wörgötter, N. Krüger, N. Pugeault, D. Calow, M. Lappe, K. Pauwels, M. V. Hulle, S. Tan, and A. Johnston, “Early cognitive vision: Using gestalt-laws for task-dependent, active image-processing,” *Natural Computing*, vol. 3, no. 3, pp. 293–321, 2004.
- [48] N. Krüger and F. Wörgötter, “Multi-modal primitives as functional models of hyper-columns and their use for contextual integration,” in *Proceedings of the 1st International Symposium on Brain, Vision and Artificial Intelligence*, vol. LNCS 3704. Springer, 2005, pp. 157–156.
- [49] C. Geib, K. Mourao, R. Petrick, N. Pugeault, M. Steedman, N. Krüger, and F. Wörgötter, “Object action complexes as an interface for planning and robot control,” *Workshop 'Toward Cognitive Humanoid Robots' at IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)*, 2006.
- [50] N. Krüger, “Signal- and symbol-based representations in computer vision,” in *Proceedings of the 1st International Symposium on Brain, Vision and Artificial Intelligence*, vol. LNCS 3704. Springer, 2005, pp. 167–176.
- [51] N. Krüger and F. Wörgötter, “Multi modal estimation of collinearity and parallelism in natural image sequences,” *Network: Computation in Neural Systems*, vol. 13, pp. 553–576, 2002.
- [52] N. Krüger, M. Lappe, and F. Wörgötter, “Biologically motivated multi-modal processing of visual primitives,” *Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, vol. 1, no. 5, pp. 417–428, 2004.
- [53] R. Wurtz and E. Kandel, *Principles of neural science (4th edition)*. McGraw Hill, 2000, ch. Central visual pathways, pp. 523–547.
- [54] A. Parker and B. Cumming, “Cortical mechanisms of binocular stereoscopic vision,” *Prog Brain Res*, vol. 134, pp. 205–16, 2001.
- [55] R. Wurtz and E. Kandel, *Principles of Neural Science (4th edition)*. McGraw Hill, 2000, ch. Perception of motion, depth and form, pp. 548–571.
- [56] J. Jones and L. Palmer, “An evaluation of the two dimensional Gabor filter model of simple receptive fields in striate cortex,” *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1223–1258, 1987.

- [57] C. Schmid and R. Mohr and C. Baukhage, "Evaluation of Interest Point Detectors," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, 2000.
- [58] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [59] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *4th Alvey Vision Conference*, 1988, pp. 147–151.
- [60] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, 1986.
- [61] J. Burns, R. Weiss, and E. Riseman, "The Non-Existence of General-Case View-Invariants," in *Geometric Invariance in Computer Vision*, J.L. Mundy and A. Zisserman, Ed. The MIT Press, 1992, pp. 120–131.
- [62] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [63] M. Brown and D. G. Lowe, "Unsupervised 3d object recognition and reconstruction in unordered datasets," in *Fifth International Conference on 3-D Digital Imaging and Modeling*, 2005, pp. 56–63.
- [64] S. Se, D. G. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2001, pp. 2051–2058.
- [65] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 02. Los Alamitos, CA, USA: IEEE Computer Society, 2004, pp. 506–513.
- [66] N. Pugeault and N. Krüger, "Multi-modal matching applied to stereo," *Proceedings of the BMVC 2003*, pp. 271–280, 2003.
- [67] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

## **Appendix H**

### **Optimal instantaneous rigid motion estimation insensitive to local minima.**

K. Pauwels and M. Van Hulle

( *Computer Vision and Image Understanding* – 2006 )

# Optimal instantaneous rigid motion estimation insensitive to local minima

Karl Pauwels \*, Marc M. Van Hulle

*K. U. Leuven, Laboratorium voor Neuro- en Psychofysiologie, Herestraat 49-bus 1021, B-3000 Leuven, Belgium*

Received 19 October 2005; accepted 4 July 2006

Available online 21 August 2006

## Abstract

A novel method is introduced for optimal estimation of rigid camera motion from instantaneous velocity measurements. The error surface associated with this problem is highly complex and existing algorithms suffer heavily from local minima. Repeated minimization with different random initializations and selection of the minimum-cost solution are a common (albeit *ad hoc*) procedure to increase the likelihood of finding the global minimum. We instead show that the optimal estimation problem can be transformed into one of arbitrary complexity, which allows for a gradual regularization of the error function. A simple reweighting scheme is presented that smoothly increases the problem complexity at each iteration. We show that the resulting method retains all the desirable properties of optimal algorithms, such as unbiasedness and minimal variance of the parameter estimates, but is substantially more robust to local minima. This robustness comes at the expense of a slightly increased computational complexity.

© 2006 Elsevier Inc. All rights reserved.

**Keywords:** Egomotion; Optic flow; Calibrated camera; Local minima; Reweighting

## 1. Introduction

The instantaneous velocity or optic flow field encountered by a moving observer contains an enormous amount of information related to the three dimensional (3D) structure of the environment and to the presence and motion of independently moving objects. Knowledge of the egomotion or self-motion of the observer is a necessary prerequisite to obtain this valuable information. Since small observer motions can have large effects on the optic flow field, it is advisable to extract the egomotion parameters from the optic flow field itself. This, however, is non-trivial and an active topic of research.

The field has matured a lot over the years and a number of ‘optimal’ algorithms (unbiased and minimal variance of the estimates) have appeared [1,2]. The error function of the optimal problem formulation is however highly nonlin-

ear and contains a large number of local minima [3,4], which renders these algorithms unreliable and hard to use in practical applications. The earlier approaches [5–8], which operate on a linearization of the problem, are no valid alternative. Compared to optimal algorithms, they are extremely sensitive to noise [1,2,9] and the estimates they provide are unsuitable, even as initializations for the optimal methods.

As an alternative to the time-consuming process of repeatedly minimizing with different, random initializations and selection of the minimum-cost solution, we propose to regularize the error function. We reformulate the problem in such a way that the complexity of the error function (the likelihood that algorithms end up in local minima) is controlled by a single parameter. We propose a reweighting scheme that gradually increases the problem complexity during the minimization, until the optimal problem formulation is obtained. We demonstrate, both in simulation and on real data, that the proposed method retains the accuracy of optimal algorithms, but is much less sensitive to local minima. On the extensive set of data investigated, these

\* Corresponding author. Fax: +32 16 345960.

E-mail addresses: [karl.pauwels@med.kuleuven.be](mailto:karl.pauwels@med.kuleuven.be) (K. Pauwels), [marc.vanhulle@med.kuleuven.be](mailto:marc.vanhulle@med.kuleuven.be) (M.M. Van Hulle).

improvements come at the cost of less than a doubling in computation time compared to previous optimal algorithms.

## 2. Problem statement

Under a static environment assumption, the motion of all points in space, relative to a coordinate system centered in the nodal point of the observer's eye, is determined by the translational velocity,  $\mathbf{t} = (t_x, t_y, t_z)^T$ , and rotational velocity,  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$ , of the moving observer. The 3D velocity,  $\mathbf{v} = (v_x, v_y, v_z)^T$ , of a point in space,  $\mathbf{x} = (x, y, z)^T$ , is then [10]

$$\mathbf{v} = -\mathbf{t} - \boldsymbol{\omega} \times \mathbf{x}. \quad (1)$$

Under perspective projection and assuming, without loss of generality, a focal length equal to unity, these 3D motion vectors are transformed into a two dimensional velocity or optic flow field. At feature location  $\mathbf{x} = (x, y, 1)^T$ , the observed flow  $\mathbf{u}(\mathbf{x}) = (u_x, u_y)^T$  equals

$$\mathbf{u}(\mathbf{x}) = d(\mathbf{x})A(\mathbf{x})\mathbf{t} + B(\mathbf{x})\boldsymbol{\omega} + \mathbf{n}(\mathbf{x}), \quad (2)$$

where

$$A(\mathbf{x}) = \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}, \quad (3)$$

$$B(\mathbf{x}) = \begin{bmatrix} xy & -1 - x^2 & y \\ 1 + y^2 & -xy & -x \end{bmatrix}. \quad (4)$$

The observed flow consists of three parts: a component due to the observer's translation (which also depends on the inverse depth  $d(\mathbf{x}) = 1/z$ ), a component due to the observer's rotation, and  $\mathbf{n}(\mathbf{x}) = (n_x, n_y)^T$ , which is assumed to be independently and identically distributed zero mean Gaussian noise. These different components are illustrated in Fig. 1. Also indicated is  $\boldsymbol{\tau}(\mathbf{x}, \mathbf{t}, 1)$ , a unit length vector orthogonal to the translational component of the flow:

$$\boldsymbol{\tau}(\mathbf{x}, \mathbf{t}, 1) = \frac{1}{\|A(\mathbf{x})\mathbf{t}\|} ([A(\mathbf{x})\mathbf{t}]_y, -[A(\mathbf{x})\mathbf{t}]_x)^T, \quad (5)$$

where  $[\mathbf{p}]_x$  and  $[\mathbf{p}]_y$  refer to the  $x$ - and  $y$ -components of the vector  $\mathbf{p}$  respectively. The meaning of the third parameter (equal to unity in Eq. (5)) is explained in Section 4. When depth is eliminated from Eq. (2), the well-known bilinear constraint [11] on translation and rotation is obtained at each location  $\mathbf{x}$

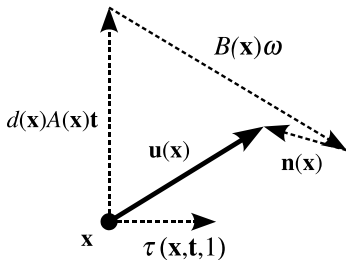


Fig. 1. Optic flow components.

$$\|A(\mathbf{x})\mathbf{t}\| \boldsymbol{\tau}(\mathbf{x}, \mathbf{t}, 1)^T (\mathbf{u}(\mathbf{x}) - B(\mathbf{x})\boldsymbol{\omega}) = 0. \quad (6)$$

This particular notation is chosen since it highlights that the constraint is weighted by  $\|A(\mathbf{x})\mathbf{t}\|$ . This weight term renders the constraints much simpler algebraically but, in the absence of prior knowledge, it is incorrect to weight the different constraints unequally. Instead, the parameters  $(\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}})$  should be estimated using the unweighted constraints [2]

$$(\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}) = \underset{\mathbf{t}, \boldsymbol{\omega}}{\operatorname{argmin}} \sum_{\mathbf{x}} [\boldsymbol{\tau}(\mathbf{x}, \mathbf{t}, 1)^T (\mathbf{u}(\mathbf{x}) - B(\mathbf{x})\boldsymbol{\omega})]^2. \quad (7)$$

These constraints represent the normalized, orthogonal deviations from the epipolar lines, and the estimates obtained from Eq. (7) minimize the least-squares image-reprojection error [4]. Since algorithms that operate on this error function obtain the most accurate parameter estimates, they are commonly referred to as 'optimal' [1,2].

## 3. Previous algorithms

A wide variety of egomotion-estimation methods have been proposed in the past. An important distinction can be made between the earlier approaches, which suffer from biased and/or widely varying estimates, and the more recent optimal algorithms.

### 3.1. Non-optimal algorithms

One of the first egomotion algorithms has been introduced by Bruss and Horn [11] and consists of a straightforward minimization of the bilinear constraints (Eq. (6)) using nonlinear optimization techniques. Heeger and Jepson (H&J) [5] have proposed a method to compute the heading (normalized translation) without iterative numerical optimization. Their linear subspace method is based on the construction of a set of constraint vectors that are independent of camera rotation. Another linear algorithm has been recently proposed by Ma et al. [6] and is conceptually similar to methods that operate on the discrete epipolar constraint. The heading estimates computed with this algorithm have been shown to be identical to those obtained with H&J but the rotation estimates are better.

The heading estimates obtained with the aforementioned algorithms are all systematically biased. Different bias correction procedures can be found in the literature. Kanatani [7] has introduced a method that subtracts an estimate of the bias from the solution. A second correction procedure has been introduced more recently by Maclean (MAC) [8] as an adaptation to H&J. Contrary to Kanatani's method, this procedure does not require an estimate of the noise variance.

### 3.2. Optimal algorithms

An optimal, nonlinear algorithm has been introduced by Chiuso et al. (CHI) [1]. This algorithm involves a sequence of fixed-point iterations where each part of the sequence

requires solving a linear least-squares problem. Chiuso et al. have proposed iterating between estimates of  $\mathbf{t}$  and  $\{d(\mathbf{x}), \omega\}$ . Since a spherical projection model has been used in their formulation and the other algorithms assume a traditional pin-hole model, we have modified the formulation and implemented the algorithm as follows. Starting from an initial heading estimate  $\mathbf{t}^{(1)}$ , a rotation estimate  $\omega^{(1)}$  is obtained as the linear least-squares solution to Eq. (7). Using both estimates, the least-squares relative inverse depth estimates are obtained at each location  $\mathbf{x}$  as

$$d^{(1)}(\mathbf{x}) = \frac{(\mathbf{u}(\mathbf{x}) - B(\mathbf{x})\omega^{(1)})^T A(\mathbf{x})\mathbf{t}^{(1)}}{\|A(\mathbf{x})\mathbf{t}^{(1)}\|^2}, \quad (8)$$

Next, the estimates  $\{d^{(1)}(\mathbf{x}), \omega^{(1)}\}$  are used to compute a new translation estimate  $\mathbf{t}^{(2)}$  as the linear least-squares solution to the system of Eq. (2). After normalization, the sequence is repeated until the estimates converge. The iterations are stopped when the magnitude of the translation update,  $\|\Delta\mathbf{t}\|$ , drops below a certain tolerance level  $\epsilon$ , which is equal to  $10^{-13}$  in all our simulations.

Zhang and Tomasi (Z&T) [2] have introduced a second optimal algorithm. By exploiting the separability of the parameters, a very fast algorithm is obtained that performs Gauss–Newton updates in  $\mathbf{t}$ . The relative inverse depth estimates  $d^{(i)}(\mathbf{x})$  are computed in the same way as CHI (Eq. (8)) but the heading and rotation estimates are updated as

$$(\Delta\mathbf{t}^{(i+1)}, \omega^{(i+1)}) = \operatorname{argmin}_{\Delta\mathbf{t}, \omega} \sum_{\mathbf{x}} [\tau(\mathbf{x}, \mathbf{t}^{(i)}, 1)^T (\mathbf{u}(\mathbf{x}) - d^{(i)}(\mathbf{x})A(\mathbf{x})\Delta\mathbf{t} - B(\mathbf{x})\omega)]^2. \quad (9)$$

Since  $\mathbf{t}$  and  $d(\mathbf{x})$  appear as a product in Eq. (2), their absolute magnitudes cannot be determined. To remove this ambiguity, the translation update is constrained to be orthogonal to the current estimate:  $(\mathbf{t}^{(i)})^T \Delta\mathbf{t}^{(i+1)} = 0$ . From Eq. (9), only the translation update is used:

$$\mathbf{t}^{(i+1)} = \mathbf{t}^{(i)} + \Delta\mathbf{t}^{(i+1)}, \quad (10)$$

the rotation estimate is recomputed as the least-squares solution to Eq. (7) (with fixed  $\mathbf{t}^{(i+1)}$ ). This way, more accurate estimates are obtained. The translation estimate is normalized to unit length only after the algorithm has converged.

#### 4. Proposed method

As mentioned in the introduction, the optimal algorithms suffer heavily from local minima. These minima are due to singularities in the unweighted error function that arise from the normalization of the bilinear constraints (Eq. (6)) by  $\|A(\mathbf{x})\mathbf{t}\|$ . As a consequence, a singularity exists for each feature where  $\mathbf{t} \propto (x, y, 1)^T$ . Under certain conditions, which are not uncommon in real-world optic flow fields, these singularities interact and influence larger regions of heading space [3,4]. Optimal algorithms initialized with a heading estimate in these regions are then likely to get trapped in a non-optimal local minimum. The weighted (bilinear) constraints on the other hand do not

suffer from these singularities and consequently fewer local minima exist. Only minima due to the so-called bas-relief ambiguity persist (for details, see [1,3]) and these are fewer in number (typically two). However, since the different features are incorrectly weighted, algorithms operating on this error function are not optimal.

We propose a novel method that arrives at optimal estimates by gradually ‘unweighting’ the bilinear constraints until the unweighted error function is obtained. The method is illustrated for Z&T but can be applied to other optimal algorithms as well. The relative inverse depth estimates are again computed using Eq. (8) but the heading and rotation updates now equal

$$(\Delta\mathbf{t}^{(i+1)}, \omega^{(i+1)}) = \operatorname{argmin}_{\Delta\mathbf{t}, \omega} \sum_{\mathbf{x}} [\tau(\mathbf{x}, \mathbf{t}^{(i)}, \rho^{(i)})^T (\mathbf{u}(\mathbf{x}) - d^{(i)}(\mathbf{x})A(\mathbf{x})\Delta\mathbf{t} - B(\mathbf{x})\omega)]^2, \quad (11)$$

where

$$\tau(\mathbf{x}, \mathbf{t}, \rho) = \frac{1}{\|A(\mathbf{x})\mathbf{t}\|^\rho} ([A(\mathbf{x})\mathbf{t}]_y, -[A(\mathbf{x})\mathbf{t}]_x)^T, \quad (12)$$

Note that the constraint weighting now depends on the value of  $\rho$ , which we define as the regularization parameter. When  $\rho$  equals zero, Eq. (11) minimizes the weighted (bilinear) constraints and few local minima will be encountered. However, when  $\rho$  equals unity, the unweighted error function is minimized (Z&T) and local minima are plentiful. The novelty of our method consists of a gradual increase of  $\rho$  (and hence of the complexity of the associated error function) from zero to unity during the Gauss–Newton iterations. Different update schemes are possible, but we use the following in all our experiments. At iteration  $i$ , the regularization parameter is updated as follows:

$$\rho^{(i)} = \min \left( 1, \rho^{(i-1)} + \lambda \left[ \frac{\log_{10} \|\Delta\mathbf{t}^{(i)}\|}{\log_{10} \epsilon} \right]^+ \right), \quad (13)$$

where  $[x]^+ = \max(x, 0)$  and  $\epsilon$  equals  $10^{-13}$  (note that  $\|\Delta\mathbf{t}\| \approx \epsilon$  at convergence). The parameter  $\lambda$ , the adaptation parameter, determines the adaptation speed and its value is set to 1/4. The choice of this parameter is discussed further in Section 5.4. Since  $\rho$  is non-decreasing and upper-bounded, the scheme is guaranteed to converge. In the remainder, we refer to the proposed regularized algorithm (the adaptation scheme from Eq. (13) applied to the heading and rotation updates from Z&T) as REG. Some typical convergence traces for both Z&T (dotted line) and REG (dashed line) are shown in Figs. 2(A) and (B), with the evolution of  $\rho$  overlaid (solid line). The traces of Fig. 2(A) have been obtained on a typical problem from Section 5.1 whereas those of Fig. 2(B) have resulted from solving a difficult problem, involving very noisy optic flow. The simple update scheme from Eq. (13) smoothly increases the regularization parameter. If the update magnitude exceeds unity,  $\rho$  is left unchanged. Otherwise,  $\rho$  is updated proportionally to the size of the update; the smaller the update (indicating that a solution is close by), the stronger  $\rho$  is



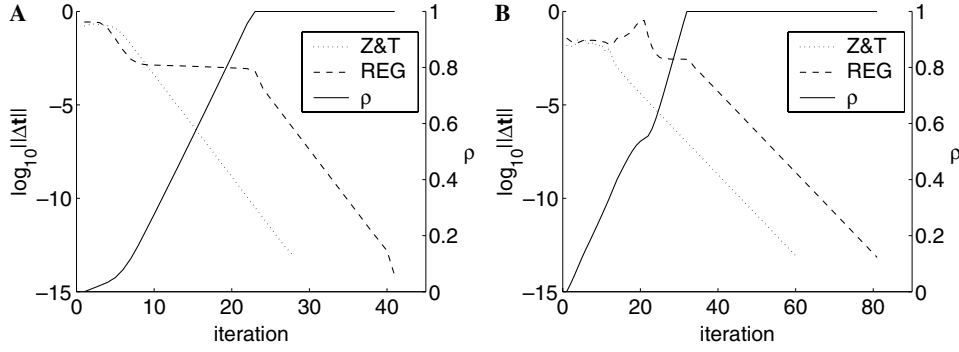


Fig. 2. Convergence traces (left y-axes) for Z&T (dotted lines) and REG (dashed lines) together with the evolution of the regularization parameter  $\rho$  (solid line, right y-axes) for two different problems; (A) a typical problem and (B) a problem with very noisy optic flow.

increased. This has a stabilizing effect on the algorithm, as exemplified by the traces of  $\rho$  and REG in Fig. 2(B) around iteration 20. As a result of the increased update magnitude at that point,  $\rho$  is increased more slowly. This in turn stabilizes the algorithm, as can be seen from the subsequent drop in the update magnitude. This increased stability warrants the slightly increased complexity of the adaptation scheme as compared to one that simply increases  $\rho$  with a fixed value at each iteration. The regularization parameter is increased until its maximum value of unity is reached. From that point on, until convergence,  $\rho$  is kept fixed and the updates are identical to those of Z&T. The convergence traces from Fig. 2 show that Z&T converges quadratically and that the regularized algorithm converges somewhat slower but still very smoothly. In the experiments performed here, the proposed method requires less than twice the number of iterations needed by Z&T (see below). Since updating  $\rho$  creates little overhead, one iteration takes an equal amount of time in both algorithms.

## 5. Experiments

In this section, the proposed method is extensively compared to some of the algorithms discussed in Section 3. First, in Section 5.1, the algorithms are compared in terms of accuracy of the parameter estimates. This evaluation involves synthetic data only and is applied to both optimal and non-optimal algorithms. Next, in Section 5.2, the proposed method's superior robustness to local minima as compared to other optimal algorithms is demonstrated. For this purpose, a synthetic problem is specifically designed so that the unweighted error function is highly complex. In Section 5.3 the proposed method's robustness is also demonstrated on the well-known real-world NASA-sequence [12]. Finally, Section 5.4 discusses the choice of the adaptation parameter  $\lambda$ .

### 5.1. Bias/variance

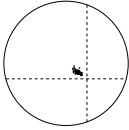
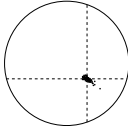
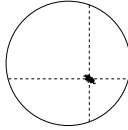
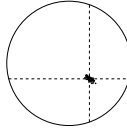
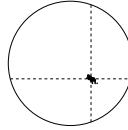
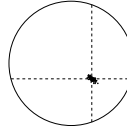
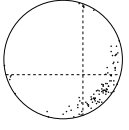
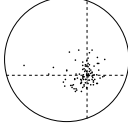
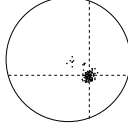
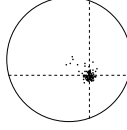
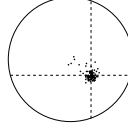
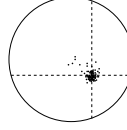
We compare H&J, MAC, CHI and Z&T to the proposed method REG in terms of the bias and variance of the heading estimates. Also included is an algorithm

identical to REG but with the regularization parameter  $\rho$  fixed to zero. This algorithm (BIL) effectively minimizes the weighted (bilinear) constraints. We use implementations provided by Tian et al. [9] for H&J, our own implementations for MAC, BIL, CHI and REG and an implementation provided by Dr. Tong Zhang for Z&T. We have not included the algorithms by Ma et al. [6] (the heading estimates of which are identical to H&J's) and by Kanatani [7] (which fails to provide unbiased estimates consistently throughout this dataset [2]). The rotation estimates are not analyzed since the bias is entirely due to heading estimation and the heading estimates can be visualized and interpreted more easily. We examine the same configuration of translation and rotation as Zhang and Tomasi [2], namely a translation and rotation direction equal to  $(4, -3, 5)^T$  and  $(-1, 2, 0.50)^T$  respectively. The rotation rate is fixed to  $0.23^\circ/\text{frame}$  and the translational magnitude is chosen so that the speeds of the translational and rotational flow components are identical in the center of the random depth cloud. In each experiment, 100 feature locations are randomly chosen and uniformly distributed over the image. The focal length is set to unity. The depth of the features is uniformly distributed between 1 and 4 units of focal length. Independently and identically distributed zero mean Gaussian noise is added to the flow vectors. The signal-to-noise ratio (SNR), defined as:  $(E\{\|\mathbf{u}\|^2\}/E\{\|\mathbf{n}\|^2\})^{1/2}$ , is varied between 10 and 30. For each algorithm, 100 trials are performed, in which the feature locations, depth and noise are randomized. For the nonlinear algorithms (BIL, CHI, Z&T and REG), 15 heading initializations, evenly spread on the unit sphere, are used and the solution with the smallest residual error is retained.

Table 1 contains the heading estimates obtained with all algorithms, for a SNR equal to 10. The field of view (FOV) is equal to  $50^\circ$  and  $150^\circ$  in the top and bottom rows respectively. The estimates are mapped to the upper hemisphere and projected onto a circle. The dashed cross marks the true heading. Example flow fields for the two conditions are shown in Fig. 3. For each algorithm and noise level, the bias, defined as the angular difference between the mean heading estimate and the actual heading, and a 95% confidence cone (measured in degrees), closely related to the



Table 1  
Heading estimates obtained with six different algorithms on 100 random trials

FOV	H&J	MAC	BIL	CHI	Z&T	REG
50°						
150°						

The FOV is equal to 50° and 150° in the top and bottom rows respectively (the SNR is equal to 10 for both). Example flow fields for these two conditions are shown in Fig. 3.

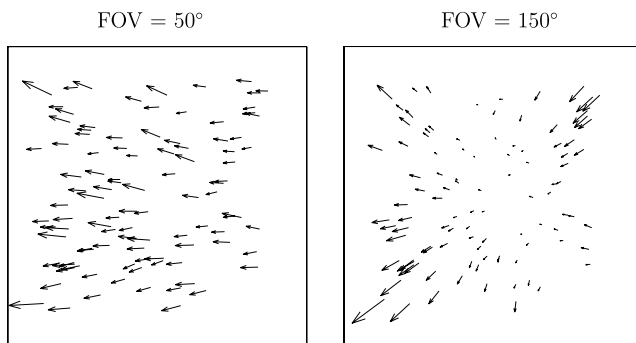


Fig. 3. Example noisy flow fields (magnified 10 times) corresponding to a FOV of 50° (left) and 150° (right). The SNR is equal to 10 in both cases.

variance of the estimates, are computed using techniques from the domain of spherical statistics [13]. Contrary to the bias/variance measures used in previous studies [1,2,9], this more sophisticated analysis clearly brings out the bias in the estimates obtained with H&J. Table 2 contains the variance measure for all algorithms, SNRs and FOVs. The value is underlined in the table if the mean heading estimate is contained within the confidence cone (unbiased). With FOV equal to 50°, this is the case for all algorithms and noise levels except, as expected, for H&J. We also see that the variance in the estimates is much

smaller for the nonlinear algorithms than for the linear ones, as observed in other studies [1,2,9]. Note that the variance for CHI, Z&T and REG is nearly identical for all configurations. However, when the constraints are weighted (BIL) the variance is about 10% larger on all occasions, which clearly demonstrates the non-optimality of this approach. Table 3 contains the median number of iterations required by the nonlinear algorithms to reach convergence for the different configurations of Table 2. The median is used since CHI and Z&T are less stable than REG and sometimes fail to converge within the maximum number of iterations (1000) allowed in our experiments. Consequently, the mean would give misleading results in favor of the proposed method. REG needs less than twice the number of iterations required by Z&T to reach convergence. The alternation steps are probably responsible for the slow convergence of CHI. Since alternation methods perform coordinate-descent, flatlining often occurs in valleys of the error surface [14]. The Gauss–Newton algorithm on the contrary, is much faster since translation and rotation are updated simultaneously.

In summary, REG performs equally well as the optimal algorithms CHI and Z&T in terms of unbiasedness and variance of the estimates and requires less than twice the number of iterations to reach convergence as compared to Z&T.

Table 2  
Radii of the 95% confidence cones (in degrees) of the heading estimates obtained with all six algorithms tested for different FOVs and SNRs

FOV	SNR	Non-optimal			Optimal		
		H&J	MAC	BIL	CHI	Z&T	REG
50°	30	0.29	<u>0.28</u>	<u>0.25</u>	<u>0.23</u>	<u>0.23</u>	<u>0.23</u>
	20	0.45	<u>0.43</u>	<u>0.38</u>	<u>0.35</u>	<u>0.35</u>	<u>0.35</u>
	10	0.86	<u>0.97</u>	<u>0.77</u>	<u>0.74</u>	<u>0.74</u>	<u>0.74</u>
150°	30	1.25	<u>1.05</u>	<u>0.45</u>	<u>0.41</u>	<u>0.41</u>	<u>0.41</u>
	20	2.57	<u>1.65</u>	<u>0.69</u>	<u>0.62</u>	<u>0.62</u>	<u>0.62</u>
	10	6.10	<u>4.13</u>	2.25	<u>2.02</u>	<u>2.03</u>	<u>2.02</u>

The value is underlined if the mean heading estimate falls within the confidence cone.

Table 3

Median number of iterations required by the nonlinear algorithms to reach convergence in the simulations of Table 2

FOV	SNR	BIL	CHI	Z&T	REG
50°	30	13	365	16	29
	20	15	368	19	32
	10	20	391	30	41
150°	30	11	118	16	33
	20	13	132	19	36
	10	16	168	26	45

## 5.2. Local minima

In the previous section we have shown that the accuracy of the proposed method is similar to that of optimal algorithms. Here, we demonstrate the greatly increased robustness to local minima that is achieved by gradually increasing the regularization parameter  $\rho$ . The error surface associated with the optimal egomotion problem is known to become flatter in a situation of lateral translation and the number of local minima increases when the feature locations are clustered together, even in the noiseless case [3]. Using this information we have constructed a particularly difficult scenario that enables us to investigate the robustness to local minima of the optimal algorithms: CHI, Z&T and REG. The egomotion consists of a translation and rotation direction equal to  $(1, 0, 0.1)^T$  and  $(0, 1, 0)^T$  respectively. The depth, translation and rotation magnitudes are chosen as in Section 5.1 and the FOV is set to  $100^\circ$ . A total of 500 features are used but, contrary to Section 5.1, they are not uniformly distributed in the image. Instead, their locations are drawn from 20 spatially distinct clusters, the centers of which are uniformly distributed over the image. The cluster centers are indicated with circles in the rightmost figure of Fig. 4. Also shown in this figure is the (subsampled and scaled) flow field used. No noise is added to the computed flow vectors. Each algorithm is run with the same 50,000 heading initializations, randomly sampled from the unit sphere, and is allowed a maximum of 1000 iterations to reach convergence. This large number of initializations allows for a detailed account of the behaviors of the algorithms over the entire heading space.

The first three figures of Fig. 4 contain the estimated headings (black circles) together with the normalized feature locations  $\mathbf{x}/\|\mathbf{x}\|$  (black dots). As before, the dashed cross marks the actual heading. It is apparent from these figures that both CHI and Z&T suffer from a large number of local minima, located near clusters of image pixels, whereas REG does not suffer from this problem at all and only finds one additional local minimum besides the global minimum (labeled A in Fig. 4). This second minimum is located near the image center and labeled B in Fig. 4. This minimum is also found by the other algorithms and is a consequence of the bas-relief ambiguity. Tech-

niques have been proposed to discriminate between these two strong minima and to quickly find the other once one is known [1]. In the remainder, we refer to local minima different from these dominant minima as undesired local minima, and to the corresponding heading initializations as undesired initializations. The fact that all undesired local minima are related to clusters of feature locations clearly indicates that they are caused by the singularities in the unweighted error function.

We repeat the experiment for different noise levels and summarize the results in Table 4: the undesired initializations (gray dots) are shown in relation to feature locations (black dots) with the number of undesired initializations underneath each instance. Besides the optimal algorithms CHI, Z&T ( $\rho = 1$ ), and the proposed method REG, we also include a number of algorithms with different, fixed, values of  $\rho$ , namely 0.9, 0.8 and 0 (BIL). Each row in Table 4 corresponds to a different noise level. In general, we observe that the number of undesired initializations increases with increasing noise. The fact that noise further increases the error surface complexity and the likelihood of convergence into a local minimum has also been observed by Oliensis [4]. As expected, the locations of these undesired initializations are related to the feature locations. It is notable that the feature clusters have a rather large spatial extent over which they exert their influence and interactions between clusters are clearly visible. The larger number of local minima of CHI is due to flatlining [14]. For all three noise levels, we see that the number of local minima gradually decreases as  $\rho$  goes to zero. When  $\rho$  equals zero, no undesired local minima are found on any occasion. This nicely illustrates how the problem complexity decreases with decreasing  $\rho$ . From the rightmost column of Table 4 it is clear that the proposed method does not suffer from undesired local minima at all, no matter the noise level. The median number of iterations for these simulations are shown in Table 5. We again see less than a doubling in computation time for REG as compared to Z&T.

Fig. 5 contains error functions of the noiseless local minima problem discussed in this section for different values of the regularization parameter  $\rho$ . The error is evaluated over an area of the image similar to Fig. 4 (rightmost). At each location  $(x, y)$  the error has been obtained by computing

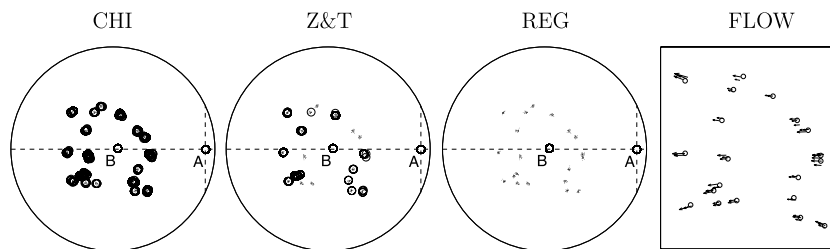
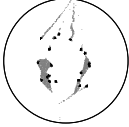
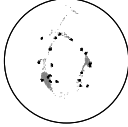
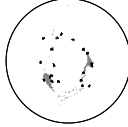
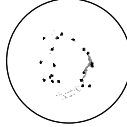
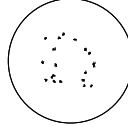
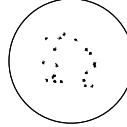

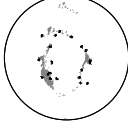
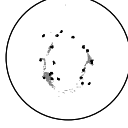
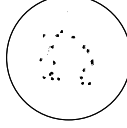
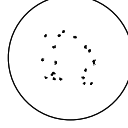
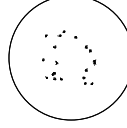
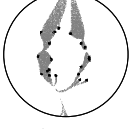
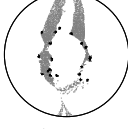


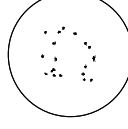
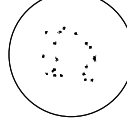


Fig. 4. Small circles in the leftmost figures correspond to heading estimates obtained with the optimal algorithms when initializing with 50,000 distinct random headings. The global minimum is labeled A and the local minimum due to the bas-relief ambiguity is labeled B. Feature locations are indicated with small black dots. The rightmost figure contains the noiseless flow field used (subsampled and magnified 10 times). In this figure, the small circles indicate the feature cluster centers.

Table 4

Undesired initializations (gray dots) in relation to feature locations (black dots) for a number of different algorithms

SNR	CHI	Z&T	$\rho = 0.9$	$\rho = 0.8$	BIL	REG
$\infty$	 2734	 1136	 827	 334	 0	 0
10	 2993	 1633	 847	 20	 0	 0
5	 7599	 7329	 2935	 2215	 0	 0

The results are shown for three noise levels. The number of undesired initializations is shown underneath each instance.

Table 5

Median number of iterations to reach convergence in the simulations of Table 4

SNR	CHI	Z&T	$\rho = 0.9$	$\rho = 0.8$	BIL	REG
$\infty$	138	7	7	7	7	7
10	144	17	17	17	17	30
5	157	32	32	32	30	45

the least-squares rotation estimate (using Eq. (7) with the current value of  $\rho$ ) assuming a candidate heading  $\mathbf{t} \propto (x, y, 1)^T$ . It is clear from this figure that the complexity of the error function smoothly increases with increasing  $\rho$ .

### 5.3. Real-world data

We repeat the analysis from the previous section on a real-world image sequence and show that the problem characteristics are not specific to our engineered data set. We use the well-known NASA-sequence [12], the center frame of which is shown in Fig. 6 (left), and compute optic flow using a phase-based algorithm [15]. Since the obtained flow field is very dense (around 50,000 vectors), we randomly select 500 flow vectors to keep the computation times reasonable. This subsampled flow field is shown in Fig. 6 (right). Next, as in Section 5.2, we run the optimal algorithms with 50,000 heading initializations, randomly sampled from the unit sphere, and allow each algorithm a maximum of 1000 iterations to converge. As before, two dominant minima are obtained for all algorithms, one of which is the global optimum (roughly forward translation). These minima are then used to identify the undesired local minima and corresponding initializations. The results are shown in Fig. 7 for CHI, Z&T

and REG. Black dots again mark the feature locations (note the small FOV) and gray dots the undesired initializations. The results are in accordance with those obtained on the synthetic datasets: REG clearly shows a superior robustness to local minima. The number of undesired initializations is 10,856 for CHI, 5018 for Z&T and only 4 for REG. The median number of iterations is 1000 for CHI, 48 for Z&T and 58 for REG. Although CHI failed to converge in more than half the trials on this very hard problem, the two dominant minima were clearly discernible. The results are consistent with those of the previous section: the reweighting scheme offers a largely increased robustness to local minima at a relatively small computational cost.

### 5.4. Choice of adaptation parameter

The parameter  $\lambda$  in the reweighting scheme (Eq. (13)) controls the speed at which the regularization parameter  $\rho$  increases during the Gauss–Newton iterations. The larger its value, the sooner  $\rho$  reaches unity and, consequently, the sooner the algorithm starts minimizing the unweighted error function. To examine the influence of the adaptation parameter on the proposed method, we ran the algorithm on the local minima problem of Section 5.2 for different values of  $\lambda$ . The SNR is fixed and equal to five on all occasions. The results are shown in Fig. 8.

Fig. 8(A) shows the number of undesired initializations as a function of  $\lambda$ . As expected, this number increases with increasing  $\lambda$ . In the limit ( $\lambda = \infty$ , which implies switching to Z&T after one iteration) 5008 undesired initializations are obtained. This is still smaller than the 7329 obtained by Z&T (see Table 4) since in the proposed reweighting scheme, the first iteration is always performed using the

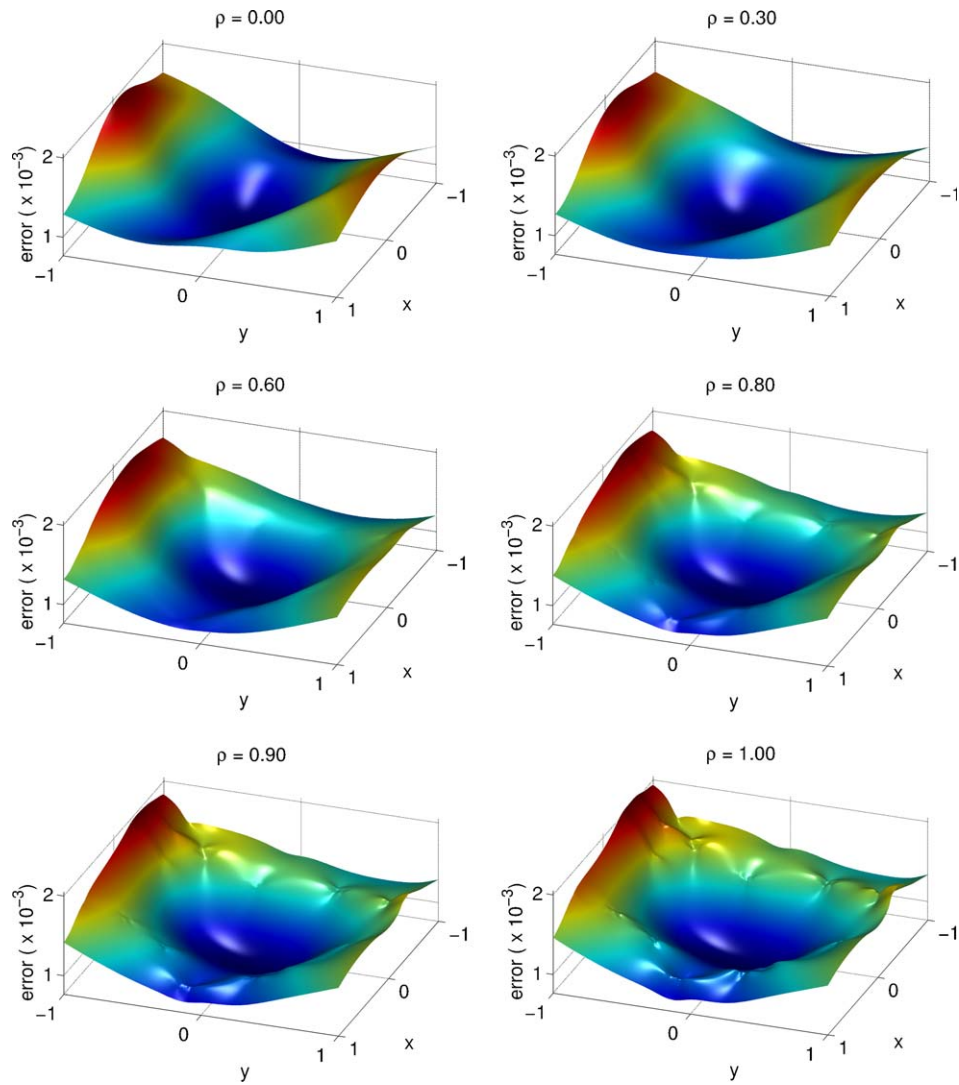


Fig. 5. Error functions of the noiseless local minima problem of Fig. 4 for different values of the regularization parameter  $\rho$ . The complexity of the error surface smoothly increases with increasing  $\rho$ .

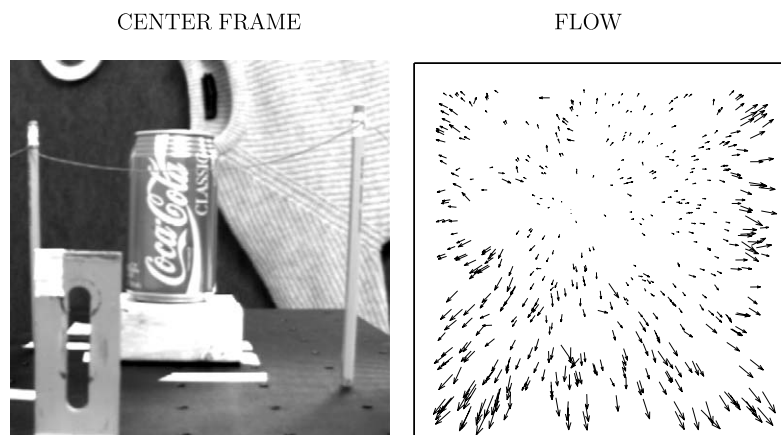


Fig. 6. The center frame of the well-known NASA-sequence (left) and 500 flow vectors (scaled) randomly selected from the complete flow field extracted from this sequence (right).

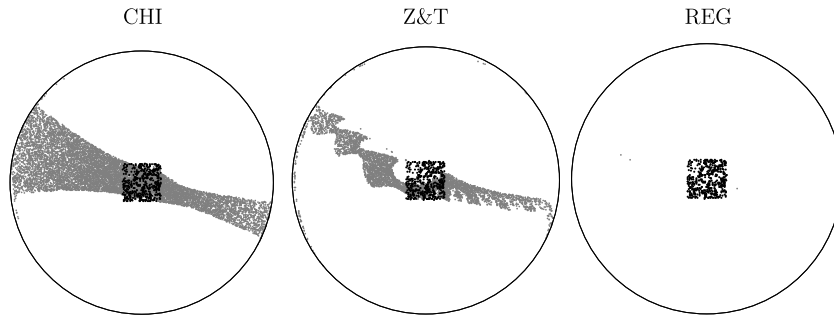


Fig. 7. Undesired initializations (gray dots) in relation to feature locations (black dots) for CHI, Z&T and REG.

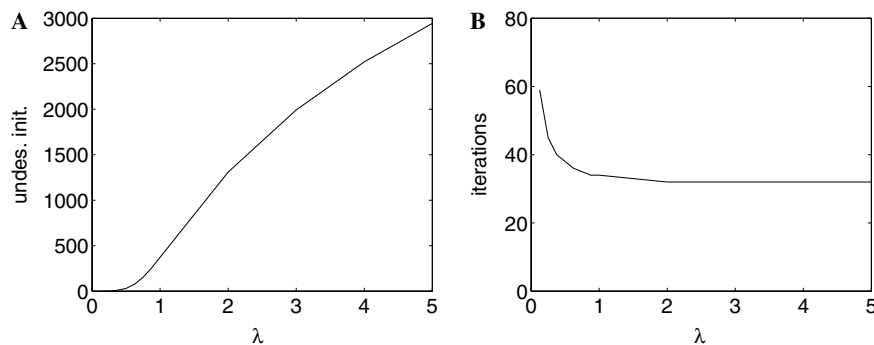


Fig. 8. Number of undesired initializations (A) and required number of iterations (B) to reach convergence on the local minima problem ( $\text{SNR} = 5$ ) as a function of the adaptation parameter  $\lambda$ .

weighted (bilinear) constraints ( $\rho = 0$ ). Fig. 8(B) contains the median number of iterations required, as a function of  $\lambda$ . Since the reweighting process slows down when  $\lambda$  is decreased, the number of iterations increases with decreasing  $\lambda$ . However, even at the smallest value of  $\lambda$  shown here ( $1/8$ ), the number of iterations is still less than twice the number required by Z&T.

We can summarize that, as long as the adaptation parameter  $\lambda$  is between zero and one, the method is relatively insensitive to its value. In this range, a reasonable tradeoff between robustness to local minima and computational requirements is obtained.

## 6. Discussion

We have presented a novel method that reduces the sensitivity to local minima of optimal egomotion-estimation algorithms by gradually increasing the problem complexity during the optimization. We have demonstrated that the local minima encountered by these algorithms are related to the feature (or feature cluster) locations and, as such, their values can be arbitrary and unrelated to the true solution. This makes these algorithms hard to use in practical applications.

As a remedy, it has been previously suggested to initialize the optimal algorithms with estimates obtained by simplified (linear) algorithms. As shown in Section 5.1 however, noise has a detrimental effect on the accuracy of linear algorithms. We have nevertheless examined this

alternative and verified that REG still outperforms Z&T in terms of robustness to local minima, even when the latter is initialized with solutions obtained by BIL (results not shown). Since the variance of all linear algorithms tested is larger than BIL, it is unlikely that their estimates will prove better initializations. An alternative way to deal with local minima is to perform multiple runs with different random initializations and retain the solution with the smallest residual. To achieve in this way the same robustness as the proposed method, a large number of runs are necessary and since our method uses fewer than twice the number of iterations required by the fastest optimal algorithm (Z&T), it is computationally more efficient.

Finally, we have shown that the proposed method behaves very similar to BIL in terms of the number of local minima found (typically two). By exploiting the relationship between these minima, the global minimum can thus be found with high certainty in only one or two runs of our method.

## Acknowledgments

Thanks to Dr. Tong Zhang and Dr. Tina Y. Tian and coworkers for providing the source code of some of the egomotion algorithms used in this paper. Thanks also to Dr. Temujin Gautama for helpful suggestions on the manuscript. Karl Pauwels and Marc M. Van Hulle are supported by the Belgian Fund for Scientific Research—Flanders (G.0248.03, G.0234.04), the Flemish Regional Ministry of

Education (Belgium) (GOA 2000/11), the Belgian Science Policy (IUAP P5/04), and the European Commission (NEST-2003-012963, IST-2002-016276, IST-2004-027017).

## References

- [1] A. Chiuso, R. Brockett, S. Soatto, Optimal structure from motion: local ambiguities and global estimates, *International Journal of Computer Vision* 39 (3) (2000) 195–228.
- [2] T. Zhang, C. Tomasi, On the consistency of instantaneous rigid motion estimation, *International Journal of Computer Vision* 46 (2002) 51–79.
- [3] T. Xiang, L. Cheong, Understanding the behavior of SFM algorithms: a geometric approach, *International Journal of Computer Vision* 51 (2) (2003) 111–137.
- [4] J. Oliensis, The least-squares error for structure from infinitesimal motion, *International Journal of Computer Vision* 61 (3) (2005) 1–41.
- [5] D. Heeger, A. Jepson, Subspace methods for recovering rigid motion I: Algorithm and implementation, *International Journal of Computer Vision* 7 (2) (1992) 95–117.
- [6] Y. Ma, J. Kosecka, S. Sastry, Linear differential algorithm for motion recovery: a geometric approach, *International Journal of Computer Vision* 36 (1) (2000) 71–89.
- [7] K. Kanatani, 3-D interpretation of optical flow by renormalization, *International Journal of Computer Vision* 11 (3) (1993) 267–282.
- [8] W. MacLean, Removal of translation bias when using subspace methods, in: *Proceedings of the Eight International Conference on Computer Vision*, IEEE Computer Society Press, Corfu, Greece, 1999, pp. 753–758.
- [9] Y. Tian, C. Tomasi, D. Heeger, Comparison of approaches to egomotion computation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, USA, 1996, pp. 315–320.
- [10] H. Longuet-Higgins, K. Prazdny, The interpretation of a moving retinal image, *Proceedings of the Royal Society of London Biology* 208 (1980) 385–397.
- [11] A. Bruss, B. Horn, Passive navigation, *Computer Graphics and Image Processing* 21 (1983) 3–20.
- [12] J. Barron, D. Fleet, S. Beauchemin, Performance of optical flow techniques, *International Journal of Computer Vision* 12 (1) (1994) 43–77.
- [13] N.I. Fisher, T. Lewis, B.J.J. Embleton, *Statistical Analysis of Spherical Data*, Cambridge University Press, Cambridge, 1987.
- [14] A. Buchanan, A. Fitzgibbon, Damped Newton algorithms for matrix factorization with missing data, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, 2005, vol. 2, pp. 316–322.
- [15] T. Gautama, M. Van Hulle, A phase-based approach to the estimation of the optical flow field using spatial filtering, *IEEE Transactions on Neural Networks* 13 (5) (2002) 1127–1136.

**Appendix I**  
***Optic Flow from Unstable Sequences containing Unconstrained Scenes through  
Local Velocity Constancy Maximization***  
K. Pauwels and M. Van Hulle  
( *BMVC 2007* )



# Optic Flow from Unstable Sequences containing Unconstrained Scenes through Local Velocity Constancy Maximization

Karl Pauwels and Marc M. Van Hulle\*

Laboratorium voor Neuro- en Psychofysiologie, K.U.Leuven, Belgium

{karl.pauwels, marc.vanhulle}@med.kuleuven.be

## Abstract

A novel stabilization method is introduced that enables the extraction of optic flow from short unstable sequences. Contrary to traditional stabilization techniques that use approximative global motion models to estimate the full camera motion, our method estimates the unstable component of the camera motion only. This allows for the use of even simpler global motion models, while at the same time extending the validity to more diverse environments, such as close scenes containing independently moving objects. The unstable component of the camera motion is derived for each frame by maximizing the temporal constancy of the local velocities over the entire short sequence. The method is embedded within a phase-based optic flow algorithm and tested on complex real-world sequences. The optic flow obtained using our technique is much denser than that extracted directly from the original sequence. The proposed method also compares favorably to a more traditional stabilization technique.

## 1 Introduction

Visual motion is a powerful sensory cue used by humans for such diverse purposes as self-motion estimation, extracting the three dimensional (3D) structure of the environment and detecting independently moving objects. This information is crucial for navigation, obstacle avoidance, *etc.* Due to the ill-posedness of the problem and external noise influences, extracting the local velocity or optic flow field from an image sequence is difficult. The quality can be greatly increased by exploiting some of the redundancy present in a short (*e.g.* five frames) image sequence. By assuming that the local velocities remain constant over this short sequence, more stable numerical differentiation techniques can be used, temporal aliasing can be reduced, and more reliable confidence measures can be computed [3, 9]. If both observer and moving objects undergo smooth motion, this velocity constancy assumption is satisfied in the majority of the scene (except in regions that become occluded during the sequence). In realistic situations however, shocks and

---

\*K.P. and M.M.V.H. are supported by the Belgian Fund for Scientific Research – Flanders (G.0248.03, G.0234.04), the Flemish Regional Ministry of Education (Belgium) (GOA 2000/11), the Belgian Science Policy (IUAP P5/04), and the European Commission (NEST-2003-012963, IST-2002-016276, IST-2004-027017).



vibrations of the vehicle or robot on which the camera is mounted result (predominantly) in fast rotational camera movements that induce large local motions over very short time spans [5]. As a result, the velocity constancy assumption is no longer valid and optic flow algorithms fail to extract meaningful motion vectors.

A typical solution is to stabilize the image sequence first. Since the unstable component of the camera motion is combined with the component that results from the self-motion, traditional stabilization techniques estimate the full camera motion and smooth it afterwards [13]. This camera motion can be decomposed into a 3D translation and a 3D rotation. The local motion field resulting from camera translation depends on the scene structure whereas that resulting from the camera rotation does not. Since both are combined, estimating camera motion in general situations is a nontrivial problem and most algorithms developed for this purpose work well in specific domains only [15]. Some stabilization techniques use *a priori* knowledge (presence of the horizon, lane markings, the road vanishing point, *etc.*) to simplify this estimation [5, 12]. This limits their applicability to situations where the required features can be reliably obtained. Most stabilization methods rely on simplified motion models instead (translation; translation, rotation and scaling; affine; quadratic; projective) and only approximate the camera motion. These models are only valid in limited scenarios (*e.g.* aerial imagery) and when they are used in more complex situations (*e.g.* driving a vehicle downtown or during vehicle turns) the stabilization algorithm typically tracks a dominant component of the background for which the model is sufficiently rich (*e.g.* the ground plane). Due to changes in the environment however, this dominant component changes also and abrupt changes in the estimated camera motion can result. For this reason, current image stabilization techniques fail when an image contains close scenes [14].

We propose a method that allows estimation of the unstable component of the camera motion only. Since this unstable component consists primarily of 3D rotations, a simple global motion model is sufficient for its estimation. Instead of *assuming* local velocity constancy, we *enforce* it and in this way exploit the fact that stable motion should result in velocity constancy locally in the majority of the scene, irrespective of the complexity of the camera motion, scene, and moving objects. By tightly integrating the stabilization with the optic flow computation, the deviations from local velocity constancy can be measured explicitly and used to estimate a global 3D rotation for each frame of the short sequence. After correcting for these rotations, the local velocity constancy and the quality of the optic flow increase greatly. Since we use only 3D rotations in the correction, the component of the flow that results from camera translation is left untouched. Consequently, the flow vectors can still be used in a variety of tasks (egomotion, structure from motion, independent motion, *etc.* can still be extracted).

The proposed stabilization technique is explained in Section 2 and extensively evaluated on two real-world sequences in Section 3. In this evaluation, the algorithm is also compared to a traditional stabilization method. Finally, concluding remarks are given in Section 4.

## 2 Image Sequence Stabilization

Our technique is embedded in an existing phase-based optic flow algorithm that we briefly present in Section 2.1. The chosen algorithm is particularly suitable for stabilization

since it relies on spatial filtering only. The proposed stabilization method is explained in Section 2.2 and a multiscale extension of the method that allows for large instabilities is discussed in Section 2.3.

## 2.1 Phase-based Optic Flow using Spatial Filtering

Fleet and Jepson [7] were the first to propose a phase-based technique for the estimation of optic flow and showed that the temporal evolution of contours of constant phase can yield a good approximation to the motion field. The proposed stabilization method centers around the phase-based optic flow algorithm by Gautama and Van Hulle [9]. The method distinguishes itself from [7] by using spatial instead of spatiotemporal filters to compute the phase, and by considering strictly local information when integrating component velocities (normal flow) into full velocities (optic flow). In an extensive comparison, similar to that from [3], the algorithm has been shown to rank among the best ones [9].

For a specific orientation, the spatial phase at pixel location  $\mathbf{x} = (x, y)$  is extracted using 2D complex Gabor filters:

$$G(\mathbf{x}, \mathbf{f}) = e^{-\|\mathbf{x}\|^2/\sigma^2} e^{i\mathbf{x}\cdot\mathbf{f}}, \quad (1)$$

with peak frequency  $\mathbf{f} = (f_x, f_y)$ . We refer to [9] for a discussion of the filterbank. The response to this oriented filter can be written as:

$$R(\mathbf{x}) = \rho(\mathbf{x}) e^{i\phi(\mathbf{x})} = C(\mathbf{x}) + iS(\mathbf{x}). \quad (2)$$

Here  $\rho(\mathbf{x}) = \sqrt{C(\mathbf{x})^2 + S(\mathbf{x})^2}$  and  $\phi(\mathbf{x}) = \arctan[S(\mathbf{x})/C(\mathbf{x})]$  are the amplitude and phase components, and  $C(\mathbf{x})$  and  $S(\mathbf{x})$  the responses of the quadrature filter pair. For every orientation  $\theta$ , the temporal phase gradient,  $\phi_{t,\theta}(\mathbf{x})$ , is computed from the temporal sequence of the spatial phase at that location,  $\phi_\theta(\mathbf{x}, t)$ , by performing a linear least-squares fit to the model (see also Fig. 1):

$$\phi_\theta(\mathbf{x}, t) = c_\theta(\mathbf{x}) + \phi_{t,\theta}(\mathbf{x})t. \quad (3)$$

A simple unwrapping technique is used to cope with the periodicity of the phase. Next, for each orientation  $\theta$  a component velocity is computed directly from  $\phi_{t,\theta}(\mathbf{x})$ :

$$\mathbf{v}_{c,\theta}(\mathbf{x}) = \frac{-\phi_{t,\theta}(\mathbf{x})}{2\pi(f_{x,\theta}^2 + f_{y,\theta}^2)} (f_{x,\theta}, f_{y,\theta}). \quad (4)$$

Note that the spatial phase gradient is substituted by the radial frequency vector. The reliability of each component velocity is measured by the mean squared error (MSE) of the linear fit:  $\sum_t (\Delta\phi_\theta(\mathbf{x}, t))^2/n$ , where  $n$  is the number of frames and:

$$\Delta\phi_\theta(\mathbf{x}, t) = (c_\theta(\mathbf{x}) + \phi_{t,\theta}(\mathbf{x})t) - \phi_\theta(\mathbf{x}, t). \quad (5)$$

Finally, provided a minimal number of reliable component velocities are obtained (threshold on the MSE), an estimate of the full velocity is computed for each pixel by integrating the valid component velocities at that pixel only:

$$\mathbf{v}^*(\mathbf{x}) = \operatorname{argmin}_{\mathbf{v}(\mathbf{x})} \sum_{\theta \in O(\mathbf{x})} \left( \|\mathbf{v}_{c,\theta}(\mathbf{x})\| - \mathbf{v}(\mathbf{x})^T \frac{\mathbf{v}_{c,\theta}(\mathbf{x})}{\|\mathbf{v}_{c,\theta}(\mathbf{x})\|} \right)^2, \quad (6)$$

where  $O(\mathbf{x})$  is the set of orientations at which valid component velocities have been obtained for pixel  $\mathbf{x}$ .

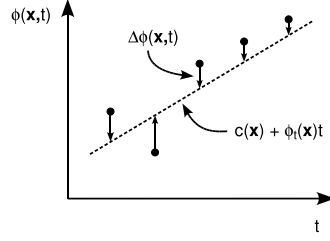


Figure 1: Temporal phase gradient linearization. For each orientation and pixel, the temporal phase gradient  $\phi_t(\mathbf{x})$  is computed by fitting a line through the spatial phases  $\phi(\mathbf{x}, t)$  computed at each frame. The proposed stabilization method aims at minimizing the deviations  $\Delta\phi(\mathbf{x}, t)$  from this estimated line by applying a global 3D stabilizing rotation  $\Delta\omega(t)$  to each frame.

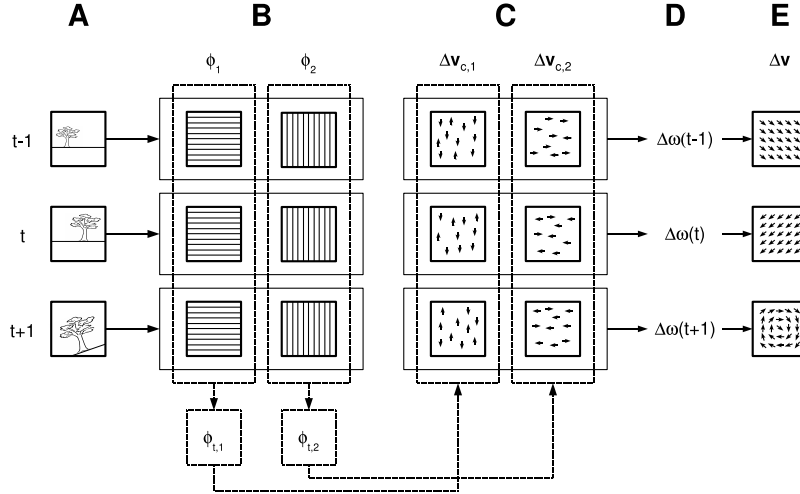


Figure 2: Stabilization overview. **(A)** A sliding window (consisting of three frames in the figure) is used to compute optic flow for the central frame  $t$ . **(B)** The spatial phase  $\phi_\theta$  is computed for each pixel, orientation  $\theta$  (two orientations in the figure) and frame  $t$ . The temporal phase gradient  $\phi_{t,\theta}$  is obtained for each pixel and orientation by fitting a linear model to the temporal sequence of the spatial phase. **(C)** The ‘unstable’ component velocities  $\Delta\mathbf{v}_{c,\theta}$  are obtained for each frame and orientation from the errors between the spatial phases and this linear model. **(D)** A 3D stabilizing rotation  $\Delta\omega(t)$  can be estimated for each frame  $t$  by integrating the ‘unstable’ component velocities over all pixels and orientations using a linear model. **(E)** These stabilizing rotations define a stabilizing full velocity field for each frame, which can be used to warp the images (or the Gabor outputs or the phases) and to obtain a stable sequence.

## 2.2 Temporal Phase Gradient Linearization

As mentioned in the introduction, the proposed method searches for a global 3D camera rotation for each frame of a short sequence that, when applied to these frames (by warping), maximizes the temporal constancy of the local velocities over the entire short sequence.

The basic idea of the method is illustrated in Fig. 1. Shown in this figure is the temporal sequence of spatial phase (after phase unwrapping) obtained at a certain pixel and for a certain orientation. A line is estimated through these points and the temporal phase gradient  $\phi_t(\mathbf{x})$  is obtained. Local velocity constancy is typically reflected in a linear evolution of the phase over time and in small errors in the line-fitting. This is clearly not the case here. The goal now is to warp the frames in such a way that the deviations from this line (small arrows) are minimized. The desired changes are computed for each pixel, orientation and frame using Eq. (5). Note that, similar to the temporal phase gradient (Eq. 4), this desired change in the spatial phase can also be interpreted as and transformed into a component velocity:

$$\Delta \mathbf{v}_{c,\theta}(\mathbf{x},t) = \frac{-\Delta \phi_\theta(\mathbf{x},t)}{2\pi(f_{x,\theta}^2 + f_{y,\theta}^2)} (f_{x,\theta}, f_{y,\theta}) . \quad (7)$$

This component velocity now reflects the local effect (orthogonal to the filter orientation) of the unstable component of the camera motion. Since we know that this component is predominantly 3D rotational [5], its estimation is straightforward. The instantaneous full velocity at pixel location  $\mathbf{x}$  that results from a 3D camera rotation,  $\boldsymbol{\omega} = (r_x, r_y, r_z)^T$ , with  $r_p$  the angular velocity around the  $p$ -axis, can be well-approximated by [1]:

$$\mathbf{v}(\mathbf{x}) = B(\mathbf{x})\boldsymbol{\omega} , \quad (8)$$

where

$$B(\mathbf{x}) = \begin{bmatrix} xy/f & -f - x^2/f & y \\ f + y^2/f & -xy/f & -x \end{bmatrix} , \quad (9)$$

and  $f$  the focal length of the camera. For component velocities we have:

$$\|\mathbf{v}_{c,\theta}(\mathbf{x})\| = (B(\mathbf{x})\boldsymbol{\omega})^T \frac{\mathbf{v}_{c,\theta}(\mathbf{x})}{\|\mathbf{v}_{c,\theta}(\mathbf{x})\|} . \quad (10)$$

On the basis of the unstable component velocities,  $\Delta \mathbf{v}_{c,\theta}(\mathbf{x},t)$ , computed at each pixel, frame and orientation we can now estimate, for each frame, the required stabilizing rotation,  $\Delta \boldsymbol{\omega}(t)$ , by solving the following linear least-squares problem:

$$\Delta \boldsymbol{\omega}^*(t) = \underset{\Delta \boldsymbol{\omega}(t)}{\operatorname{argmin}} \sum_{\mathbf{x},\theta} \left[ \|\Delta \mathbf{v}_{c,\theta}(\mathbf{x},t)\| - (B(\mathbf{x})\Delta \boldsymbol{\omega}(t))^T \frac{\Delta \mathbf{v}_{c,\theta}(\mathbf{x},t)}{\|\Delta \mathbf{v}_{c,\theta}(\mathbf{x},t)\|} \right]^2 . \quad (11)$$

Once the stabilizing rotations are found, they are used to correct the sequence and the optic flow is recomputed. The corrections can be done by warping the images or, more efficiently, the Gabor filter outputs (Eq. 2). An overview of the complete stabilization procedure is provided in Figure 2.

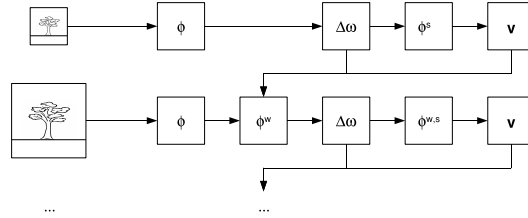


Figure 3: Multiscale stabilization.

Note that not all deviations from linearity in Fig. 1 result from instabilities. Other disturbing factors are image noise, phase singularities, motions exceeding the filter range, *etc.* These latter errors are however much weaker correlated compared to those resulting from the instabilities. Due to the sheer volume of available measurements, robust and precise rotation estimates can still be obtained.

An important limitation of the method discussed in this section is that the magnitude of the effect of the unstable camera motion component has to be within the range of the Gabor filters. To extend this range and to enable the method to also detect and correct large rotational shocks, the stabilization technique can be embedded in a coarse-to-fine multiscale implementation of the optic flow algorithm. This is the subject of the next section.

### 2.3 Multiscale Extension

Due to phase periodicity, phase-based techniques can only detect shifts that do not exceed half the filter wavelength. To extend this range, a coarse-to-fine control strategy can be used [8]. An efficient solution involves the use of an image pyramid, in which the image resolution is halved at each level. By applying the original filters to each level of the pyramid, the detectable range of shifts is doubled at each level. The control strategy starts at the lowest resolution and uses optic flow estimates obtained there to warp the images at the next higher resolution so that the estimated motion is removed [4]. The residual motion is then within the range of the filters applied at that level.

The optic flow algorithm we use is particularly suitable for this warping strategy since it uses strictly local information. In a similar fashion as in Section 2.2, we do not warp the images themselves but rather the filter outputs. In our implementation, only optic flow vectors that can be computed reliably at the highest resolution are retained. In other words, if the refinement made at the highest resolution to a lower resolution estimate (that was reliable at that lower resolution) is unreliable, the flow vector is discarded and not included in the density counts of the next section. In this way, overly smooth flow fields are avoided.

Figure 3 contains a schematic overview of the coarse-to-fine control strategy used in the proposed stabilization technique. The procedure starts at the lowest resolution. The spatial phase  $\phi$  is computed at this level and the stabilizing rotations  $\Delta\omega$  are estimated as explained in Section 2.2. These rotations are then used to warp the filter outputs and compute the stable phase  $\phi^s$  and stable full velocities  $\mathbf{v}$ . The stabilizing rotations and full velocities are then transformed (multiplied by two) to the next scale and the filter

seq	single scale			multiscale		
	ORG	TRA	PGL	ORG	TRA	PGL
city	<u>31.5</u>	<u>37.1</u>	<u>40.1</u>	<u>37.9</u>	<u>44.8</u>	<u>52.2</u>
mway	<u>22.6</u>	26.2	25.8	<u>32.0</u>	<u>32.8</u>	<u>37.1</u>

Table 1: Average flow field density (in percent).

outputs at that level are warped to compensate for the effects of these motions. Next, the stabilization procedure is applied to this motion-compensated phase  $\phi^w$  and a refinement of the stabilizing rotations is obtained. The filter outputs are then rewarped to incorporate this refinement  $\phi^{w,s}$  and the residual full velocities are computed. Finally, the updated rotations and full velocities are propagated to the next level and the procedure is repeated until the original resolution is obtained.

### 3 Results

We evaluate the proposed Phase Gradient Linearization method (PGL) in terms of the optic flow density (the percentage of reliable flow vectors) obtained before and after stabilization. A full velocity is considered reliable if the MSE of the linear fit (Eq. 3) does not exceed 0.01 for at least five (out of 11) of the component velocities used in its estimation. Five frames are used in the computation and three scales are used in the multiscale implementation of the algorithm. We also evaluate the optic flow density after stabilization with a popular alternative stabilization technique. This technique (TRA) estimates a 2D translation globally by matching the images as a whole [2]. We use the normalized cross correlation measure for reliable matching. Subpixel accuracy is obtained by refining this estimate with a gradient-based technique [10]. Central differences are used to estimate the spatial derivatives. This combined procedure enables high-precision image registration. A linearization procedure similar to that shown in Fig. 1 is used to correct the individual 2D translation estimates and to render the estimated camera motion constant over the short sequence (a unique transformation is obtained by fixing the central frame).

Both techniques are applied to two complex real-world driving sequences, recorded in different environments. The sequences have been recorded with a camera rigidly installed behind the front shield of a moving car<sup>1</sup>. The first one, *city*, contains close scenes and relatively small vehicle velocities whereas the second sequence, *mway*, involves larger vehicle speeds and also larger destabilizing motions. Moving objects are present in both sequences. An example image of each sequence, together with the optic flow computed for these frames is shown in Fig. 4. It is clear from this figure that the flow computed after stabilization with PGL looks very similar to that computed without stabilization (ORG), except for the greatly increased density. This is because the stabilization procedure averages out the instabilities over the entire short sequence.

The complete sequences each consist of  $\pm 450$  frames of  $320 \times 256$  pixels, and the obtained optic flow densities are summarized in Table 1. A two-way ANOVA and Tukey multiple comparison test [11] have been used to assess the significance of all individual

<sup>1</sup>Courtesy of Dr. Norbert Krüger, Aalborg University Copenhagen, and HELLA Hueck KG, Lippstadt.

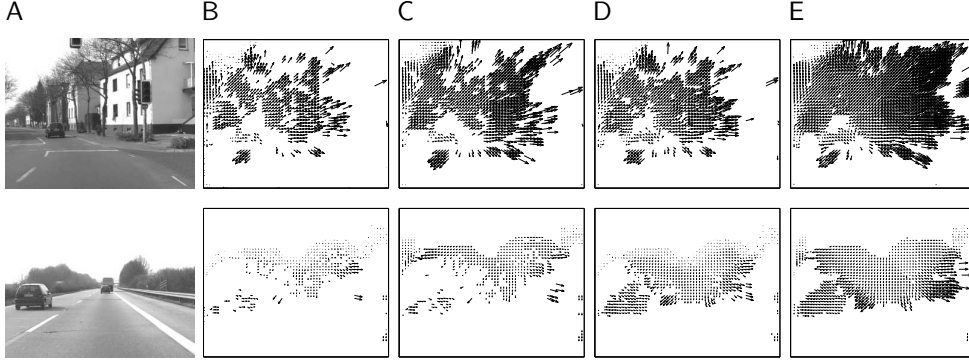


Figure 4: Example images (A) and flow fields (B–E) obtained on the *city* (top row) and *mway* (bottom row) sequence without stabilization using (B) single scale and (C) multi-scale optic flow, and with the proposed stabilization using (D) single scale and (E) multi-scale optic flow. All flow fields have been subsampled and scaled 10 times.

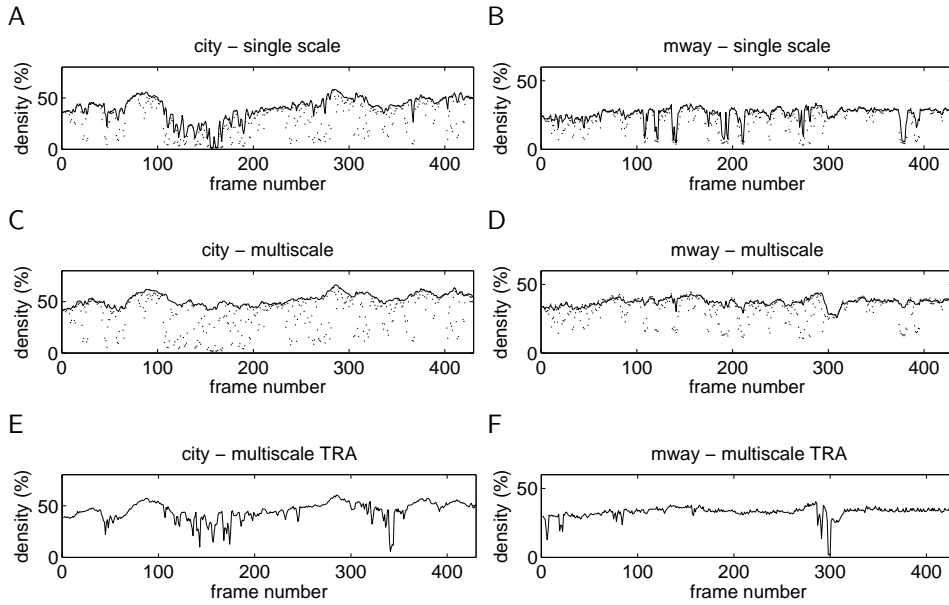


Figure 5: Optic flow field density. (A–D) Results obtained without stabilization (black dots) and after stabilization with the proposed method (solid line) over the entire *city* (left) and *mway* (right) sequences. The first and second row correspond to the results obtained with the single and multiscale algorithm respectively. (E,F) Results obtained with the alternative stabilization method (TRA) on both sequences using the multiscale implementation.

pairwise differences in mean density at the joint significance level of 0.05. The mean density is underlined in the table if all pairwise differences in which the respective algorithm occurs are significant. This analysis is repeated for each combination of sequence and control strategy (single scale/multiscale). The multiscale strategy improves the density on all occasions. The TRA stabilization technique significantly improves the density as compared to the original sequences, but the proposed method achieves far better results in general and in the multiscale scenario in particular.

Figure 5 shows the improvements obtained with PGL in more detail. In this figure the optic flow density is shown as a function of frame number for the entire sequences. In Figs. 5(A–D), the densities obtained without stabilization are shown as black dots and those obtained after stabilization with PGL as solid lines. We can already see significant improvements in the single scale case but the technique fails at certain frames (*e.g.* around frame 150) in *city* and at various locations in *mway*. The multiscale stabilization overcomes this problem, which clearly shows that large unstable motions are present here (the multiscale results without stabilization are as bad as the single scale at these frames). In the multiscale case, an almost constant density stream of optic flow is obtained over the entire sequences after stabilization. For completeness, the density obtained with TRA is shown in Figs. 5(E,F). Due to the prevalence of close scenes in *city*, the procedure fails often. Better results are obtained on *mway*, but the stabilization is still unreliable and the density is often smaller than that obtained without stabilization.

To make sure that the weaker results of TRA are not from its inability to model rotations around the line of sight, we have repeated the simulations with the proposed method, but now using a simple 2D translation model in Eq. (11). The results were not significantly different from those obtained with the full 3D rotation model. This could be either because instabilities do not result in rotations around the line of sight in these sequences or because of inaccuracies resulting from rotating (warping) the filter outputs. Since rotations change the orientations, refiltering or a more efficient framework such as steerable filters [6] may be required to further improve the precision. The latter allows for changes in orientation without refiltering. This is a subject of further investigations.

## 4 Conclusion

We have proposed a novel stabilization technique that does not require estimation of the full camera motion but enables a direct estimation of the unstable component of the camera motion. This is achieved through a maximization of the temporal constancy of the local velocities. The method is computationally efficient as it involves linear systems and simple transformations, the result of which can be computed without time-consuming re-filtering. Although we use a global motion model of similar complexity, we achieve significant increases in reliable optic flow density on real-world sequences as compared to a traditional stabilization technique. It is true that evermore complex global motion models can be used to more accurately model the camera motion in alternative techniques, but this will be at the cost of efficiency, stability, and simplicity. Our method on the other hand is simple and valid in the most general of scenes, those where the distance to the scene is small, the range of depths within the scene is large, and moving objects are present. By using only 3D rotations in the stabilization, the information in the optic flow that relates to the depths of the scene is left undisturbed.



## References

- [1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, 1985.
- [2] S. Balakirsky and R. Chellappa. Performance characterization of image stabilization algorithms. *Real-Time Imaging*, 12(2):297–313, 1996.
- [3] J.L. Barron, D.J. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
- [4] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, 1992.
- [5] Z. Duric and A. Rosenfeld. Shooting a smooth video with a shaky camera. *Machine Vision and Applications*, 13(5–6):303–313, 2003.
- [6] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.
- [7] D.J. Fleet and A.D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.
- [8] D.J. Fleet, A.D. Jepson, and M.R.M. Jenkin. Phase-based disparity measurement. *CVGIP-Image Understanding*, 53(2):198–210, 1991.
- [9] T. Gautama and M.M. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Trans. Neural Networks*, 13(5):1127–1136, 2002.
- [10] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.
- [11] J.C. Hsu. *Multiple Comparisons: Theory and Methods*. Chapman & Hall, London, 1996.
- [12] Y.M. Liang, H.R. Tyan, S.L. Chang, H.Y.M. Liao, and S.W. Chen. Video stabilization for a camcorder mounted on a moving vehicle. *IEEE Transactions on Vehicular Technology*, 53(6):1636–1648, 2004.
- [13] C. Morimoto and R. Chellappa. Fast electronic digital image stabilization for off-road navigation. *Real-Time Imaging*, 2(5):285–296, 1996.
- [14] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):694–711, 2006.
- [15] T. Xiang and L.F. Cheong. Understanding the behavior of SFM algorithms: A geometric approach. *International Journal of Computer Vision*, 51(2):111–137, 2003.

## **Appendix J**

### **Fixation as a Mechanism for Stabilization of Short Image Sequences**

K. Pauwels, M. Lappe, and M. Van Hulle

*( International Journal of Computer Vision 72(1), 2007 )*



# Fixation as a Mechanism for Stabilization of Short Image Sequences

KARL PAUWELS, MARKUS LAPPE AND MARC M. VAN HULLE

*Laboratorium voor Neuro-en Psychofysiologie, K.U. Leuven, Belgium. M. Lappe is with the  
Psychologisches Institut II, Westf. Wilhelms-Universität, Münster, Germany*

*Received March 30, 2005; Revised December 29, 2005; Accepted February 3, 2006*

*First online version published in June, 2006*

**Abstract.** A novel method is introduced for the stabilization of short image sequences. Stabilization is achieved by means of fixation of the central image region using a variable window size block matching method. When applied to a sliding temporal window, the stabilization improves the performance of standard optic flow techniques. Due to the unique choice of fixation as the main stabilization mechanism, the proposed method not only increases the flow field density but renders certain global structural properties of the flow fields more predictable as well. This in turn is advantageous for egomotion computation.

## 1. Introduction

Visual motion is one of the more important sensory cues that are used by humans to guide behavior or to navigate a dynamical environment. The instantaneous velocity or optic flow field contains a tremendous amount of information related to the self-motion of the observer, the three dimensional (3D) structure of the environment, and the presence and motion of independently moving objects. Extracting this velocity field from the temporal evolution of image intensity values is a highly complex and ill-posed problem. In order to obtain unique solutions, a variety of assumptions have been used to constrain the problem. One important assumption, adopted by many optic flow algorithms proposed in the literature, states that the local velocities remain constant over a short time span (Barron et al., 1994). If this assumption holds, multiple frames can be used in the estimation process. This allows for the application of more stable numerical differentiation techniques, the reduction of temporal aliasing (Barron et al., 1994) or the extraction of more reliable confidence measures (Gautama and Van Hulle, 2002). When both observer and moving objects undergo smooth motion, this velocity constancy assumption is valid (except

at motion boundaries). In realistic situations however, the computation of optic flow has to cope with undesired motion of the camera due to shocks or vibrations of the vehicle or robot on which it is mounted. These perturbations typically manifest themselves as fast, rotational camera movements (Duric and Rosenfeld, 2003) that induce large local motions over very short time spans (Giachetti et al., 1998). Consequently, the assumption of locally constant velocities is often violated. A possible solution is to use optic flow algorithms that do not make this assumption (Giachetti et al., 1998), such as correlation-based matching techniques. Since the performance and reliability of these techniques on stable sequences, is typically much lower than those of a differential or phase-based approach (Barron et al., 1994), a better solution is to stabilize the image sequence first. After stabilization, the velocity constancy assumption is met more closely, and consequently, a differential or phase-based approach can be used to compute optic flow.

### 1.1. Stabilization

Image sequence stabilization is defined as the process of modifying an image sequence from a moving or

jittering camera so that it appears stable or stationary (Balakirsky and Chellappa, 1996). Traditional stabilization techniques estimate the camera motion first and use it to render the sequence stable. This egomotion or rigid self-motion of the camera can be decomposed into a 3D translation and a 3D rotation. Due to motion parallax, the translational motion field depends on the scene structure, while the rotational motion field is fully determined by the camera parameters only. The superposition of these two components can result in complicated motion fields. Although much progress has been made to date, extracting the camera motion from such optic flow fields is nontrivial and most algorithms perform well only in specific domains (Xiang and Cheong, 2003). A distinction can be made between 2D and 3D techniques for electronic image stabilization. The former proceed by fitting an affine model to all motion in the sequence (Morimoto and Chellappa, 1996). This renders them very efficient but limits their validity to scenes with minimal depth variation (*e.g.* aerial images). In contrast, 3D stabilization techniques operate on the camera rotation only and consequently do account for a rich scene structure. This approach is effective since in normal situations (such as driving or walking), the effects of unwanted translations are negligible compared to the effects of unwanted rotations (Duric and Rosenfeld, 2003). These 3D techniques stabilize by de-rotating the frames, in this way generating a translation-only sequence (Irani et al., 1997), or by temporally smoothing the rotational component of the camera motion (Duric and Rosenfeld, 2003). Note that this involves estimating the rotation in the presence of general motion, with all its associated difficulties and ambiguities.

### 1.2. Fixation

The stabilization strategy adopted by humans and primates is quite different: motion in the fovea or central, high-resolution part of the retina is nullified by means of eye movements. These gaze stabilization eye movements use vestibular, proprioceptive, or visual signals to achieve this task (Lappe and Hoffmann, 2000). For the present work we use the term fixation to describe the effect of such eye movements, that is to hold the gaze direction towards the same environmental point through time (Daniilidis, 1997; Fermüller and Aloimonos, 1993; Lappe and Rauschecker, 1995). Contrary to other 3D stabilization techniques, fixation does not require estimation of the rotational component of

self-motion and is hence much simpler. Instead, on the basis of foveal motion only, a compensatory, 3D rotation (eye movement) is determined and superposed on the motion field. Since rotational jitter acts on every part of the image or retina, this procedure effectively removes its effects.

The stabilization method introduced here is very similar and aims at fixating the central image region in a short image sequence. A novel variable window size block matching procedure, that allows for joint feature selection and feature tracking, enables the fixation point to remain at this location. By using a correlation-based matching technique, velocity constancy is not required at this stage. Since the method specifically aims at improving the computation of optic flow by increasing the temporal velocity constancy, the length of the sequence is determined by the temporal support required by the optic flow algorithm. The choice of fixation as the mechanism for stabilization not only renders the procedure relatively simple (as compared to other 3D stabilization methods) but has a number of additional advantages as well. First of all, it is well known that fixation reduces the number of parameters that determine the egomotion from five (two for the heading or translation direction and three for the rotation) to four (Aloimonos et al., 1987). The reason for this is that the horizontal and vertical rotations that stabilize the fixation point (*e.g.* the image center) are fully determined by the (relative) depth of that point and the current translation. This observation has been exploited in numerous algorithms (Daniilidis, 1997; Fermüller and Aloimonos, 1993; Lappe and Rauschecker, 1995; Taalebinezhad, 1992) that compute egomotion from optic or normal flow. A second advantage is related to the global structure of flow fields obtained during fixation. Typically, during fixation and self-motion, the singular point of the optic flow field is near the center of the visual field (fovea) (Lappe and Rauschecker, 1995). Therefore, this central area contains many different local motion directions that are important for the analysis of the flow field. In contrast, in the periphery speed and homogeneity of the flow increase with distance from the center (*cf.* center flow field in Fig. 1B). This allows spatial averaging over a larger scale without losing too much information about the local motion directions (Lappe, 1996). In other words, fixation results in a consolidation of information near the fovea. These global properties are quite robust to scene changes, heading changes, and small fixational errors and are therefore a good basis for the development

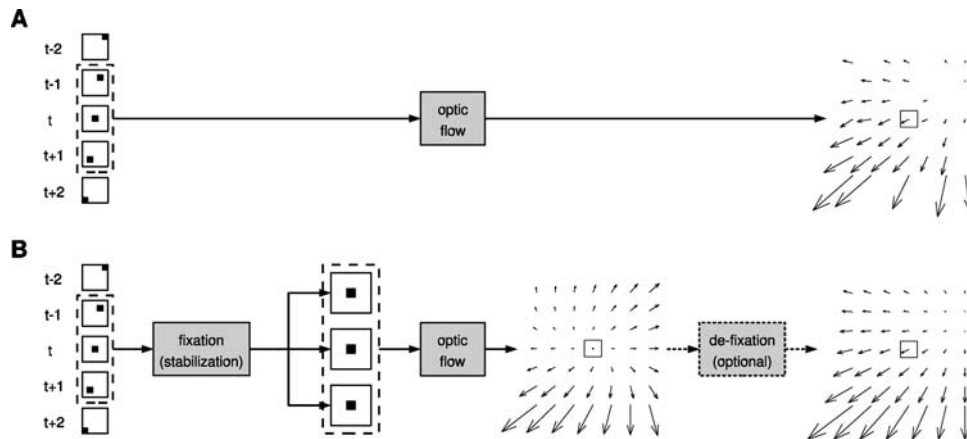
of space-variant filtering techniques that improve egomotion computation (Calow et al., 2006). Furthermore, they can directly benefit the computation of optic flow itself. By fixating prior to flow estimation, the parameters for the estimation (*e.g.* filter sizes) can be predicted to scale with eccentricity, to a certain extent. In this way, the performance of single-scale algorithms can be improved and the increased complexity of, and computational resources required by multi-scale algorithms avoided.

A number of artificial fixation systems have been proposed in the past. Most of these systems are active (they control the camera motion) and employ feedback to fixate a region of interest (Fermüller and Aloimonos, 1993). Besides being active, they differ from the proposed method in that these regions need to be selected either manually or by means of ‘interest point detectors’. A passive tracking/fixation system is discussed in (Taalebinezhad, 1992). This latter method however fixates two images to simplify egomotion estimation and is not suitable for image sequence stabilization.

## 2. Proposed Method

In this section we give a brief overview of the proposed stabilization method and explain in what way it alters classical optic flow computation. Figure 1 illustrates both the classical (A) and proposed (B) approach graphically.

Typical approaches to compute optic flow for each frame of a long image sequence involve the use of a sliding temporal window. A short window, the length of which depends on the temporal support required by the optic flow algorithm, is moved over the sequence one frame at a time and the instantaneous velocity field is computed for the central frame of the window (Section 2.3). This window is marked by the dashed boxes in Figure 1 and contains three frames in this example. As illustrated in Figure 1A, when optic flow is extracted from these frames directly, the obtained flow field is often sparse and noisy. The proposed stabilization method operates on the images in these short windows, and optic flow is computed only after all images within the sliding window are stabilized. Stabilization consists of a simulated fixation (Section 2.1) of the central part of the short image sequence. The feature that is at the image center at time  $t$  is marked by the small filled squares in Figure 1. Fixation involves detecting and tracking this feature over the current temporal window (Section 2.2). After stabilization, its location remains fixed in the image center. Next, optic flow is computed on this ‘fixated’ image sequence. Due to the stabilizing effect of this fixation, the resulting flow field is typically less noisy and denser than the one computed directly on the original image sequence. As discussed in the introduction, certain global structural properties of the fixated flow field differ from those of the original flow field. Note how the fixation has added a rotational curl to the center flow field in Figure 1B and rendered the image center (indicated with the small



*Figure 1.* Classical optic flow computation (A) and the proposed method (B). The dashed box marks the sliding temporal window used in computing the optic flow at time  $t$ . Without stabilization, the flow field is sparse and noisy (right flow field in A). The small filled squares mark the location of the feature that is at the image center at time  $t$ . After fixation, this feature is motionless in the warped images (B). Note how a rotational curl is present in the flow field computed on the stabilized images. An optional de-fixation step can transform the flow field into one that more closely resembles the flow field computed on the original sequence.

square) motion-free. Although not necessary for most purposes, certain applications require flow fields that more closely resemble those computed on the original image sequence. For this reason, the stabilization procedure contains an optional de-fixation step (Section 2.4) that removes the rotational stabilization effects from the optic flow field. The resulting flow field is shown to the right in Fig. 1B and looks very similar to the original one from Fig. 1A, except that the former is less noisy and denser.

### 2.1. Image Sequence Stabilization

Similar to other active and passive systems that exploit foveal representations (Daniilidis, 1997; Fermüller and Aloimonos, 1993), the optical image center (the intersection of the optical axis with the image) is chosen as the fixation point in our method. This is similar to the biological case in the sense that it corresponds to the direction of gaze. Although the location of the fixation point does not affect the generality of the method, choosing the center has certain advantages, such as allowing for the same amount of warping in all directions. Keeping this location fixed renders the procedure conceptually simple and yields more stable global structural properties of the flow field (Section 4.4), which in turn can be exploited efficiently by hardware architectures.

Fixation is achieved by means of simulated 3D rotations around the  $x$ - and  $y$ -axes of the observer-centered coordinate system<sup>1</sup>. Although relevant in the context of stabilization,  $z$ -axis rotations are not considered here (see also Section 2.2), without loss of generality of the fixation procedure. Figure 2 illustrates the stabilization method for an example sequence consisting of five frames. To transform the sequence into a fixated sequence, *i.e.* a sequence in which the central image part is motion-free, the central part of the middle frame (the ‘template window’, indicated by the small solid square) needs to be localized in all frames of the sequence. A straightforward way to achieve this tracking

would be to block match the central part of frame 3 directly to all other frames. However, to allow for gradual texture changes and to limit the size of the search windows (dashed squares), tracking is performed iteratively in our method. To match backward from frame 3 to frame 1, the texture in the center square of frame 3 is first matched to the area within the search window in frame 2. The obtained displacement (arrow in frame 2) is used to move the search window in frame 1 and the texture found in frame 2 (small square) is then matched to this search window. A similar procedure is followed to match forward to frame 5. These displacements uniquely determine a 3D rotation for each frame that warps the texture most similar to the central texture of the middle frame to the center of the respective frame.

As an example, we determine the rotation for frame 1 from Fig. 2. The center coordinates of the template window in frame 1 equal:  $(x_1, y_1) = \mathbf{d}_{32} + \mathbf{d}_{21}$ . Since the stabilization operates on short temporal windows, a velocity-based scheme yields a reasonable approximation of the 3D rotation (Adiv, 1985). In this scheme, the instantaneous velocity  $(\dot{x}, \dot{y})$  of image point  $(x, y)$  resulting from the camera rotation  $(\omega_x, \omega_y, \omega_z)$  equals:

$$\dot{x} = \omega_x \frac{xy}{f} - \omega_y \left( f + \frac{x^2}{f} \right) + \omega_z y \quad (1)$$

$$\dot{y} = \omega_x \left( f + \frac{y^2}{f} \right) - \omega_y \frac{xy}{f} - \omega_z x, \quad (2)$$

where  $f$  is the focal length of the camera. Consequently, the compensatory 3D rotation that warps  $(x_1, y_1)$  to the center pixel  $(0, 0)$  should result in the following motion vector at  $(x_1, y_1)$ :

$$\dot{x}_1 = -x_1 \quad (3)$$

$$\dot{y}_1 = -y_1. \quad (4)$$

Since we only consider  $x$ - and  $y$ -axis rotations in the stabilization, a unique compensatory 3D rotation

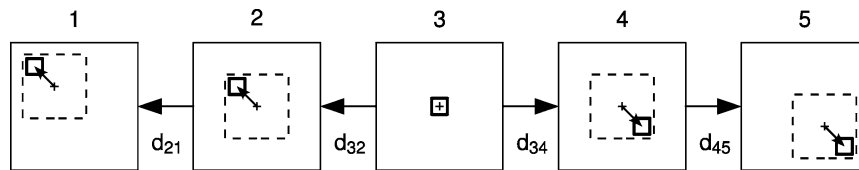


Figure 2. Stabilization by means of fixation. The central image region of frame 3 is backward and forward tracked to frames 1 and 5 respectively. In this way, the individual displacements  $\mathbf{d}_{ij}$ , denoting the movement of the feature from frame  $i$  to frame  $j$ , are determined.

satisfies this requirement:

$$(\omega_x, \omega_y, \omega_z) = \left( -\frac{y_1 f}{f^2 + x_1^2 + y_1^2}, \frac{x_1 f}{f^2 + x_1^2 + y_1^2}, 0 \right). \quad (5)$$

This rotation is now used to warp every pixel  $(x, y)$  in frame 1 according to Eqs. (1) and (2). Cubic convolution interpolation (Keys, 1981) is used to perform these warps with subpixel accuracy.

After warping each frame (except the middle frame) according to the stabilizing rotations, the central part of the image sequence is motion-free. Note that the interframe rotations are not necessarily identical. In case there is a need to reconstruct the original flow fields, these rotations must be averaged in the de-fixation step (Section 2.4).

## 2.2. Variable Window Size Matching

As discussed in the previous section, the stabilization method requires tracking the central region of the middle frame over the short image sequence. All matching is performed using the normalized cross correlation method (Lewis, 1995). Since the location of the fixation point is set in advance, the use of a fixed window size at this location can result in a textureless template window. This is contrary to most approaches to feature tracking which use interest point detectors to first localize regions in the image that contain certain types of textures or features that simplify matching. Fixed-window block-matching techniques are then typically used to track these regions over different frames. Although the proposed method is not allowed to change the location of the fixation point, the size of the template window can be chosen freely. To ensure the general applicability of the method, the window size should be increased in the absence of texture or in ambiguous situations due to a repetitive pattern. In the context of stereo matching, Kanade and Okutomi (1994) proposed an adaptive window method that optimally balances between signal-to-noise ratio or intensity variation maximization and projective distortion (due to variations in the depth of scene points) minimization. This technique is however unable to deal with repetitive patterns. It is very important to take such ambiguities into account, since they can result in large estimated displacements that may deteriorate the subsequent computation of optic flow. A possible approach to detect spurious matches is to analyze the cross-correlation surface in terms of its peakedness (Anandan, 1989). However, such analysis

requires a set of relatively arbitrary thresholds, so that its reliability can be called into question (Barron et al., 1994).

On the basis of two heuristics, we propose a simple and robust matching algorithm that effectively combines feature selection and feature matching. The first heuristic is founded on the observation that when a repetitive pattern is accidentally matched to a wrong instance, it is unlikely that an identical displacement is obtained when the matching is repeated with a slightly larger window. The heuristic consists of increasing the window size until two successive matches result in the same displacement vector. This yields excellent results in most cases and typically results in very small template windows. However, there still remain situations where the procedure is confused by strong repetitive patterns. Most matching techniques validate local matches by means of global constraints inherent to the problem (*e.g.* stereo or rigid body motion). A constraint we can employ here is the following: if we track a feature over three consecutive frames 1, 2, and 3, the displacements from frame 1 to 2 ( $\mathbf{d}_{12}$ ) and from 2 to 3 ( $\mathbf{d}_{23}$ ) should add up to the displacement obtained when directly matching frame 1 to frame 3 ( $\mathbf{d}_{12} + \mathbf{d}_{23} = \mathbf{d}_{13}$ ). The combination of these heuristics results in the following matching algorithm for matching frame 1 to frame 2, using frames 1, 2, and 3:

### INITIALIZE

template window size  $w = 0$   
 search window size  $s = 0$   
 displacements  $\mathbf{d}_{12}^0, \mathbf{d}_{23}^0, \mathbf{d}_{13}^0 = \text{NaN}$   
 iteration  $i = 0$

### DO

$w = w + 10$  ;  $s = w + 50$  ;  $i = i + 1$   
 match frame 1 to frame 2  $\rightarrow \mathbf{d}_{12}^i$   
 match frame 2 to frame 3  $\rightarrow \mathbf{d}_{23}^i$   
 match frame 1 to frame 3  $\rightarrow \mathbf{d}_{13}^i$

### UNTIL

$\mathbf{d}_{12}^{i-1} = \mathbf{d}_{12}^i$  ;  $\mathbf{d}_{23}^{i-1} = \mathbf{d}_{23}^i$  ;  $\mathbf{d}_{13}^{i-1} = \mathbf{d}_{13}^i$   
 $\mathbf{d}_{13}^i = \mathbf{d}_{12}^i + \mathbf{d}_{23}^i$

In the next frame, matching is performed using the constraint  $\mathbf{d}_{23} + \mathbf{d}_{34} = \mathbf{d}_{24}$ . This is continued until the complete short sequence is stabilized. In a single step of the algorithm, the same template and search window sizes are used for all three matches. Note that this simple algorithm requires only two parameters:

the increase in the template window size after each iteration and the size of the search window, relative to the template window size.

Since this matching component is a crucial part of the proposed method, we apply an additional subpixel refinement step after all pixelwise displacements are estimated. Assuming that the above-mentioned matching procedure correctly computes the integer parts of the displacements, we further refine these estimates by computing the least-squares fit to the gradient constraint equation (Horn and Schunck, 1981). The subpixel displacement  $(s_x, s_y)$  is chosen that minimizes the constraint deviation over the smallest template window  $\Omega$  that yields the correct (pixelwise) displacement estimates:

$$\sum_{\mathbf{x} \in \Omega} \left( I_x(\mathbf{x}, t)s_x + I_y(\mathbf{x}, t)s_y + I_t(\mathbf{x}, t) \right)^2, \quad (6)$$

where  $I_p(\mathbf{x}, t)$  is the partial derivative of the image intensity function to parameter  $p$  at pixel  $\mathbf{x} = (x, y)$  and time  $t$ . These partial derivatives are approximated by forward differences (after compensating for the pixelwise motion). Instead of Eq. (6), a more complex motion model that also incorporates rotations around the line of sight ( $z$ -axis) could be used at this stage. This has not been included here for two reasons. First of all, a richer model might reduce the accuracy of the displacement estimates. Secondly, contrary to the determination of the fixational rotation, which is restricted to a small area surrounding the fixation point, the extraction of  $z$ -axis rotation can exploit information located anywhere in the image. Consequently, instead of increasing the model complexity at the template window, an even more sophisticated procedure, not restricted to this window, is more appropriate.

In certain situations, it is possible that relatively large template windows are necessary and that the stabilized sequence no longer fixates exactly on the image center. Imperfections in the tracking, however, only result in imperfect fixation, but not in incorrect flow or egomotion computation, since the performed warps are known and can be used to reconstruct the original flow (see Section 2.4). Therefore, only algorithms that build on a perfectly fixated flow field are affected by this.

### 2.3. Optic Flow

To demonstrate the consistency of our results, we use two fundamentally different optic flow algorithms. The

first algorithm is the well-accepted differential-based algorithm by Lucas and Kanade (1981) (LUC). As suggested in Barron et al. (1994), the image sequence is first smoothed with a spatiotemporal Gaussian filter with a standard deviation of 1.5 pixels-frames before computing the derivatives. We use image sequences of length 13 to have sufficient temporal support. The second algorithm is a more recent phase-based algorithm by Gautama and Van Hulle (2002) (GAU). This algorithm uses spatial filtering to compute phase components of oriented filters at every time frame. The temporal phase gradient is estimated from this sequence of phase components using linear regression. Finally, an intersection-of-constraints step extracts the full velocity from the component velocities. The resulting optic flow fields have been shown to be much denser and more accurate than those obtained with LUC (Gautama and Van Hulle, 2002). For this algorithm, we use the parameters suggested in Gautama and Van Hulle (2002). No pre-smoothing is required here and the algorithm uses only five frames.

### 2.4. De-fixation

Figure 3 contains flow fields for an example frame of one of the sequences (Section 3) used in the analyses. The top and bottom row flow fields have been extracted using LUC and GAU respectively. The optic flow in the center column has been computed directly on the original sequence whereas the left column flow has been computed after fixation. When comparing these two columns, it is clear that the flow fields can look very different. A comparison of these two flow fields is important for the validation of the stabilization method. Even though it is not required for the computation of the translational egomotion parameters and the subsequent recovery of structure from motion, certain applications may also prefer operating on flow fields that more closely resemble the flow fields computed on the original sequence, or may require knowledge of the true rotational egomotion parameters. To achieve these goals, the fixating rotation needs to be determined and the original flow reconstructed by ‘de-fixating’ the stabilized flow fields, *i.e.* removing the effects of this fixating rotation. Since the interframe rotations that stabilize the short image sequence are not necessarily identical, de-fixation requires their summarization into a single rotation.

The most sensible way to proceed is by averaging the individual rotations in the same way as the optic flow



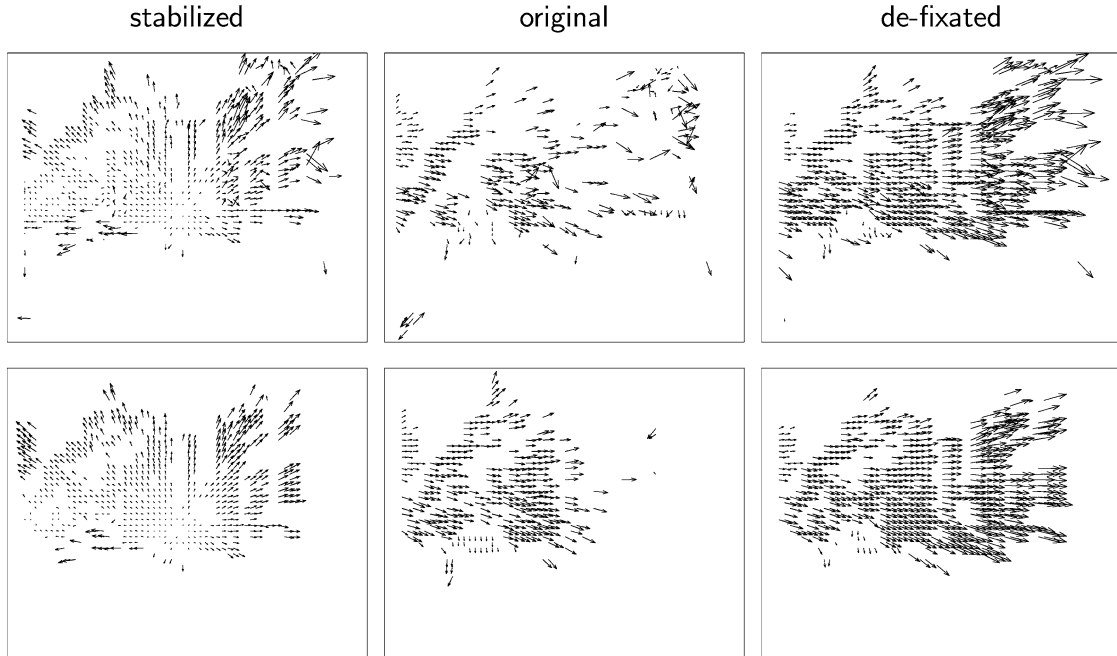


Figure 3. Flow fields computed for the *city3* frame shown in Figure 4. The flow has been computed using LUC (top row) and GAU (bottom row). The left and middle columns contain the flow fields computed respectively with and without stabilization. The right column contains the stabilized flow fields after removal of the stabilizing rotation. All flow fields have been subsampled and scaled 10 times.

algorithm averages the temporal information over the sequence. For the phase-based algorithm, all five frames are equally weighted, so a simple averaging of the four interframe rotations yields the best results. In our Lucas and Kanade implementation 13 frames are spatiotemporally convolved with a Gaussian of standard deviation 1.5 pixels-frames, and the five central frames are retained. On the basis of these five frames, derivatives are computed with four-point central differences by convolution with the mask:  $\frac{1}{12}(-1, 8, 0, -8, 1)$ . We apply a similar transformation to compute the average rotation. In this way, each individual rotation influences the computation of the average rotation in a similar way as the respective frame influences the computation of the temporal derivatives. This is achieved by first convolving the interframe rotations with the same Gaussian used in the flow computation, and then computing the average rotation as the weighted average of the four central interframe rotations, with weights equal to  $\frac{1}{18}(1, 8, 8, 1)$ .

The de-fixation procedure has been applied to the flow fields in the left column of Fig. 3 and the results are shown in the right column. It is clear that for both algorithms the de-fixated flow fields very closely resemble the ones computed on the original sequences

(except that the former are denser and less noisy). In conclusion, we can see that, although stabilization can arbitrarily change the inter-frame rotations over a short sequence, it is still possible to extract a single fixating rotation and to reconstruct the flow, as corresponding directly to the original sequence.

### 3. Sequences

Three real-world driving sequences are used in the analyses. The sequences have been recorded with a camera rigidly installed behind the front shield of a moving car<sup>2</sup>. All sequences are 18 seconds long and contain 450 frames at a resolution of  $638 \times 508$  pixels. The sequences contain a wide variety of inner-city driving situations. An example frame from each sequence is shown in Figure 4. Stabilizing these sequences is nontrivial, as the scenes exhibit large depth variability and stable features (*e.g.* the horizon) are lacking. The sequences differ with respect to the curvature of the trajectory, illumination conditions, and the overall condition of the road. The latter directly relates to camera jitter. Note that even though the camera is fixed relative to the car, this does not imply a constant heading. When



Figure 4. Example frames from the three sequences used. All sequences consist of 450 frames and contain a wide variety of driving situations and illumination conditions.

the car moves along curves or overtakes other cars, the heading strongly deviates from a forward translation. Although only driving sequences are used in the evaluation, no characteristics specific to this kind of sequences (such as the high speed or the presence of a road) are exploited by the method. Consequently, the method is applicable in more general situations involving self-motion (*e.g.* walking in natural scenes).

## 4. Results

In this section, the effects of stabilization on the computed optic flow are investigated by comparing density and global structure of the optic flow fields computed before and after stabilization. To show the merits of our proposed fixation approach, two other stabilization methods are included in the comparison as well. Both techniques are explained next.

### 4.1. Alternative Stabilization Techniques

The first technique (TRA) registers two images by estimating a 2D translation globally, using the whole images. This mechanism is typically used in electronic stabilization systems of commercial cameras. In our implementation, images are matched by applying the normalized cross correlation technique to the entire images. Although time-consuming, this is effective.

The second technique (PHC) is more sophisticated and estimates the best-fitting affine transformation (2D translation, 2D rotation, and scale) between two images. As mentioned in the introduction, for scenes with minimal depth variation this transformation largely accounts for the camera motion. The affine transformation is found by performing phase correlation, both in

the original space (to find the 2D translation) and in log-polar space (to find the rotation and scale) (Reddy and Chatterji, 1996).

Both registration techniques are applied in the stabilization framework explained in Section 2.1. In a similar fashion as the proposed method, all frames of the short sequence are matched to the center frame. Only consecutive frames are registered and the estimated transformations are accumulated. A similar procedure to the one described in Section 2.4 is used to compute the average transformations for TRA and PHC, which can be used to reconstruct the original flow fields from the stabilized if desired.

### 4.2. Optic Flow Reliability Measures

When evaluating the density and global structure of the optic flow fields, only reliable flow vectors are considered. Two different reliability measures are computed for each flow vector and only if both agree, the flow vector is retained.

A first measure of reliability is provided by the optic flow algorithms themselves. For LUC, a velocity estimate is retained if the least-squares matrix used in solving the gradient constraint equation (a weighted version of Eq. 6) is invertible (Barron et al., 1994). GAU considers a full velocity estimate to be reliable if at least five component velocities are used in its determination (a total of 11 component velocities are computed at each location). A component velocity is rejected if the corresponding filter pair's phase information is not linear over the short sequence.

In addition to this first measure, a second reliability measure is computed. This measure, the image reconstruction quality, is independent of the flow

algorithm and allows for flow field transformations (e.g. de-fixation) before evaluation. Given the optic flow vector  $(\dot{x}, \dot{y})$  at location  $(x, y)$  and time instant  $t$ , we define the image reconstruction quality as the normalized correlation between the intensity values of small windows centered at  $(x, y)$  and  $(x + \dot{x}, y + \dot{y})$  in frames  $t$  and  $t + 1$  respectively. A flow vector is considered reliable when this correlation exceeds 0.9. The correlation is computed over windows of size  $15 \times 15$  pixels and cubic interpolation is used to achieve sub-pixel accuracy in the comparison. Measures based on the reconstruction quality have been shown to yield adequate performance in evaluating flow vector quality (Lin and Barron, 1995).

#### 4.3. Optic Flow Field Density

The flow field density is the number of reliable flow vectors divided by the number of pixels. For the original flow fields, the image reconstruction quality is evaluated directly on the original images. For the stabilized flow fields, the average effect of the stabilizing transformations is first removed from the flow fields, using the de-fixation procedure for FIX and similar reconstruction procedures for TRA and PHC. In this way, the reconstruction quality is also evaluated on the original images. This allows for a more direct comparison between the different flow fields. Note that this also validates that the stabilization and reconstruction procedures preserve the dynamic aspects of the sequence. Table 1 contains the average density of reliable flow vectors before and after stabilization for all algorithms on all three sequences. Since the density varies widely across frames, the frame index is included as a factor in a two-way ANOVA. Using a Tukey multiple-comparison test (Hsu, 1996), the significance of all individual pairwise differences in mean density

is assessed at the joint significance level of 0.05. The mean density is underlined in the table if all pairwise differences in which the respective algorithm occurs are significant. This analysis is repeated for each combination of sequence and optic flow algorithm.

For the proposed method FIX, stabilization results in a significant increase in flow density as compared to the original sequence on all occasions. For optic flow algorithm LUC, we see that FIX performs better than TRA but is outperformed by PHC on all sequences. This is due to the estimation of scale by the registration component of PHC, which results in smaller displacements in the stabilized sequences on average (see also Figure 5). As a consequence of this, the number of flow vectors that are within the acceptable magnitude bounds of the single-scale flow algorithm increases. Even though this is also the case for optic flow algorithm GAU, a very different result is obtained. Here PHC and TRA perform much worse than FIX, and the obtained densities are not significantly different from those computed on the original sequence (they are even smaller for *city1*). The reason for this weak performance is that both PHC and TRA are whole-image techniques that lack a tracking component. In other words, they do not guarantee that the same features are matched over the entire short sequence, as does the proposed method. This is not a problem if the model employed by the registration technique is a good approximation of the camera movement, but due to the rich scene structure of the sequences used, this is not the case here. Although PHC yields good results when registering two frames, inconsistencies occur in longer sequences. As a result, the local velocities no longer remain constant and the estimates are rejected by the reliability measure of GAU. It is clear from the results that this effect strongly outweighs the advantages resulting from the average magnitude reduction. This effect is weaker for LUC since this optic flow algorithm strongly smooths

Table 1. Average flow field density (in percent) obtained on the original sequence (ORG) and after stabilization using 2D translation (TRA), phase correlation (PHC), and fixation (FIX). The mean density is underlined if all pairwise differences in which the respective algorithm occurs are significant. For each combination of sequence and optic flow algorithm, the joint significance level of all pairwise differences is 0.05.

seq	LUC				GAU			
	ORG	TRA	PHC	FIX	ORG	TRA	PHC	FIX
city1	<u>21.3</u>	<u>21.6</u>	<u>26.8</u>	<u>22.0</u>	<u>24.2</u>	18.4	19.4	<u>27.6</u>
city2	21.5	21.8	<u>24.7</u>	<u>23.0</u>	22.1	20.4	21.2	<u>27.0</u>
city3	<u>15.9</u>	<u>17.6</u>	<u>23.2</u>	<u>19.3</u>	14.8	15.1	15.6	<u>23.0</u>

the sequences before estimating the gradients. As a consequence, the reliability measure is less sensitive to small inaccuracies. This smoothing however leads to less accurate flow estimates (Gautama and Van Hulle, 2002).

#### 4.4. Global Flow Field Structure

As discussed in the introduction, fixation renders certain global flow field properties more predictable. In particular, speed and homogeneity of the flow vectors tend to increase with distance from the fixation point. The speed effects can be quantified by evaluating the mean and standard deviation of the flow vector magnitude as a function of eccentricity (the fixation point is the image center). These values are computed by averaging, for each frame, the flow vector magnitudes within specific eccentricity rings and summarizing these values over all sequences. The results are shown in Figure 5. The mean and standard deviation of the magnitudes are shown in the left and right columns respectively. The results are qualitatively similar for both optic flow algorithms.

For the original sequence (dashed lines) and TRA (dotted lines), the mean magnitude increases slightly with eccentricity and the standard deviation remains

large throughout, as compared to the other algorithms. As expected, for PHC (dash-dotted lines) the mean magnitudes are strongly reduced at all eccentricities. The standard deviation is also much smaller. This renders the magnitude of the velocity vectors well-predictable, but less so near the fovea.

Finally, the results for the proposed method FIX (solid lines) show a very strong upward trend in the mean motion magnitudes and a small standard deviation throughout. Note that this does not necessarily imply that after stabilization, the flow field is purely translational with focus of expansion in the center (see *e.g.* the stabilized flow field in Figure 1B). Differences with PHC occur near the fovea, where FIX results in smaller magnitudes and standard deviations, and at large eccentricities, where the mean magnitudes are larger for FIX.

In conclusion, both the proposed stabilization by fixation and PHC render the global structure of the optic flow fields more predictable. The structure imposed by the proposed method is however much more pronounced. As can be expected from a fixation-based system, the image is very well stabilized near the center. In this way, static image processing in general becomes much easier at this location. For a system that has to perform many tasks at once, this may be very important.

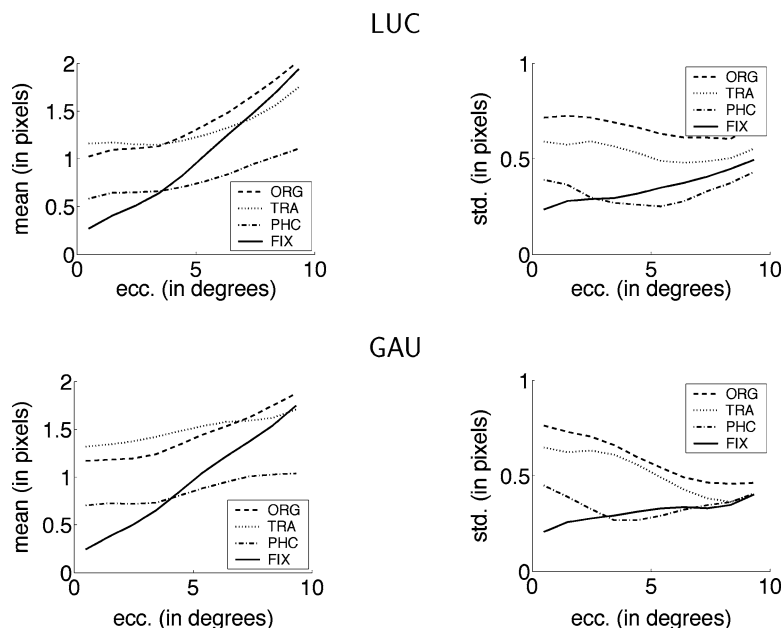


Figure 5. Mean (left column) and standard deviation (right column) of the optic flow vector magnitude as a function of eccentricity with and without stabilization. The results have been summarized over all sequences and are shown in the top and bottom row for LUC and GAU respectively.

## 5. Conclusion

The proposed method achieves stabilization by fixating short image sequences. After stabilization, optic flow computation is greatly facilitated. It has been argued that this processing order and the techniques developed to achieve it, can provide important advantages that enable a more robust extraction of behaviorally relevant information, such as camera motion, structure from motion, and independent motion. First, the improved flow density allows for a more accurate egomotion estimation using egomotion algorithms that are proven consistent (Zhang and Tomasi, 2002). Second, during fixation, the number of parameters required to describe the egomotion is reduced from five to four. Last, fixation renders the global flow field structure better predictable and results in a consolidation of information near the fovea, which is advantageous for the application of optimized noise filtering and/or data compression techniques. This increased structural consistency also enables one to define, in advance, sensible space-variant parameters for single-scale optic flow algorithms.

Although possible extensions related to the compensation of  $z$ -axis rotation have not yet been included in the algorithm, significant quantitative improvements of stabilization with respect to optic flow density and global flow structure have been demonstrated. In an extensive comparison with established stabilization procedures, it has been shown that sequences stabilized with the proposed method are better conditioned for highly accurate optic flow algorithms. Furthermore, the global structure of the resulting flow estimates is much more pronounced.

## Acknowledgment

K.P. and M.M.V.H. are supported by research grants received from the Belgian Fund for Scientific Research—Flanders (G.0248.03 and G.0234.04), the Flemish Regional Ministry of Education (Belgium) (GOA 2000/11), the Interuniversity Attraction Poles Programme—Belgian Science Policy (IUAP P5/04), and the European Commission (IST-2001-32114, IST-2002-001917 and NEST-2003-012963). M.L. is supported by the German Science Foundation DFG LA-952/2 and LA-952/3, the German Federal Ministry of Education and Research BioFuture Prize, and the EC Projects EcoVison, Eurokinesis, and Drivisco.

## Notes

1. In this coordinate system the  $x$ -axis is horizontal, the  $y$ -axis vertical and the  $z$ -axis coincides with the line of sight. The origin corresponds to the optical center of the camera.
2. All sequences have been recorded in the context of the ECOVISION project. Courtesy of Dr. Norbert Krüger, Aalborg University Copenhagen, and HELLA Hueck KG, Lippstadt.

## References

- Adiv, G. 1985. Determining Three-dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401.
- Aloimonos, Y., Weiss, I., and Bandyopadhyay, A. 1987. Active Vision. *International Journal of Computer Vision*, 1(4), 333–356.
- Anandan, P. 1989. A Computational Framework and an Algorithm for the Measurement of Visual-motion. *International Journal of Computer Vision*, 2(3):283–310.
- Balakirsky, S. and Chellappa, R. 1996. Performance Characterization of Image Stabilization Algorithms. *Real-Time Imaging*, 12(2):297–313.
- Barron, J., Fleet, D., and Beauchemin, S. 1994. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):43–77.
- Calow, D., Krüger, N., Wörgötter F., and Lappe, M. 2006. Biologically Motivated Space-variant Filtering for Robust Optic Flow Processing. *Network: Computation in Neural Systems (in press)*.
- Daniilidis, K. 1997. Fixation Simplifies 3D Motion Estimation. *Computer Vision and Image Understanding*, 68(2):158–169.
- Duric, Z. and Rosenfeld, A. 2003. Shooting a Smooth Video with a Shaky Camera. *Machine Vision and Applications*, 13(5–6):303–313.
- Fermüller, C. and Aloimonos, Y. 1993. The Role of Fixation in Visual Motion Analysis. *International Journal of Computer Vision*, 11(2):165–186.
- Gautama, T. and Van Hulle, M. 2002. A Phase-based Approach to the Estimation of the Optical Flow Field Using Spatial Filtering. *IEEE Transactions on Neural Networks*, 13(5):1127–1136.
- Giachetti, A., Campani, M., and Torre, V. 1998. The Use of Optical Flow for Road Navigation. *IEEE Transactions on Robotics and Automation*, 14(1):34–48.
- Horn, B. and Schunck, B. 1981. Determining Optical Flow. *Artificial Intelligence*, 17(1–3):185–203.
- Hsu, J. 1996. *Multiple Comparisons: Theory and Methods*. London: Chapman & Hall.
- Irani, M., Rousso, B., and Peleg, S. 1997. Recovery of Ego-Motion Using Region Alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):268–272.
- Kanade, T. and Okutomi, M. 1994. A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9): 920–932.
- Keys, R. 1981. Cubic Convolution Interpolation for Digital Image Processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160.
- Lappe, M. 1996. Functional Consequences of an Integration of

- Motion and Stereopsis in Area MT of Monkey Extrastriate Visual Cortex. *Neural Computation*, 8:1449–1461.
- Lappe, M. and Hoffmann, K. 2000. Optic Flow and Eye Movements. In: M. Lappe (ed.): *Neuronal Processing of Optic Flow*. Academic Press, 29–47.
- Lappe, M. and Rauschecker, J. 1995. Motion Anisotropies and Heading Detection. *Biological Cybernetics*, 72:261–277.
- Lewis, J. 1995. Fast Template Matching. In: *Vision Interface*, 120–123.
- Lin, T. and Barron, J. 1995. Image Reconstruction Error for Optical Flow. In: C. Archibald and P. Kwok (eds.): *Research in Computer and Robot Vision*. Singapore: World Scientific Publishing Co., 269–290.
- Lucas, B. and Kanade, T. 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. In: *Proc. DARPA Image Understanding Workshop*, 121–130.
- Morimoto, C. and Chellappa, R. 1996. Fast Electronic Digital Image Stabilization for Off-road Navigation. *Real-Time Imaging*, 2(5):285–296.
- Reddy, B. and Chatterji, B. 1996. An FFT-based Technique for Translation, Rotation, and Scale-invariant Image Registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271.
- Taalebinezhad, M. 1992. Direct Recovery of Motion and Shape in the General Case by Fixation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):847–853.
- Xiang, T. and Cheong, L. 2003. Understanding the Behavior of SFM Algorithms: A Geometric Approach. *International Journal of Computer Vision*, 51(2):111–137.
- Zhang, T. and Tomasi, C. 2002. On the Consistency of Instantaneous Rigid Motion Estimation. *International Journal of Computer Vision*, 46(1):51–79.

**Appendix K**  
**Structural Visual Events**  
N. Pugeault, F. Wörgötter and N. Krüger  
( *Technical Report 2007 – 1* )

# Structured Visual Events

Nicolas Pugeault, Norbert Krüeger and Florentin Wörgötter

January 23, 2007



Title	Structured Visual Events
	Copyright © 2007 Nicolas Pugeault, Norbert Krüeger and Florentin Wörgötter. All rights reserved.
Author(s)	Nicolas Pugeault, Norbert Krüeger and Florentin Wörgötter
Publication History	

# 1 Introduction

The human visual system is efficient at grouping together visual information that belongs to the same objects, regardless of noise and ambiguity. Salient objects immediately ‘pop out’ of the visual environment. Gestalt psychologists suggested that this emergence of some coherent sub-parts of the scene is driven by a certain number of rules, also called *Gestalt Laws*. These laws stated that certain regularities lead the visual system to group together visual information that would otherwise be, from a local signal viewpoint, distinct. Such laws included, e.g., proximity, good continuation, similarity and symmetry. Striking demonstrations of such a bias in the human visual system exist in the form of so-called *visual illusions*: e.g. the Kanisza triangle, where an illusory triangle is strongly perceived. There has been discussions that such laws might be originated by statistical properties in natural images. This was later demonstrated by [18, 8, 12]. In [7] a statistical approach was used to extract close contours. The statistical part was mainly concerned with the pairwise grouping of local edge pixels. [4] proposed a complementary statistical scheme to extract global groups from such information. We believe that such an approach can be extended to extract salient image structures without prior assumption on the scene witnessed or the objects that constitute it, and we propose to call those *Structured Visual Events (SVE)*. The primordial sort of structural SVE is a contour, and in its simplest form, the line. As discussed in [4], the likelihood for accidental alignment of edge pixels (or alternatively local edge-like features) is decreasing with the square of the size of the contour. Such SVE correspond to the Gestalt law of *Good Continuation*, and therefore we propose that more SVE could be inferred according to the other aforementioned laws.

In the present work we will consider the following regularities:

- Parallelism
- Coplanarity (in space, described in [15]).
- Similarity (co-colority, described in [15]).
- Good continuation (described in [25]).

All of these regularities are defined in 3D space, or alternatively across stereo in both images — see [15] for a detailed description.

We will propose a simple scheme to extract salient locations in the images, salient in the sense of a statistical oddity that is likely to correspond to an object in the scene. We will use in conjunction the above-mentioned relations to segment the visual world into Structured Visual Events and background. Note that the segmentation of visual scenes is a difficult problem, that found some satisfying solutions in the limited case of foreground/background segmentation, but that is otherwise unsolved. [9]

## 2 Visual primitives

Numerous feature detectors exist in the literature (see [23] for a review). Each feature based approach can be divided into an interest point detector (e.g. [13, 3]) and a descriptor describing a local patch of the image at this location, that can be based on histograms (e.g. [5, 23]), spatial frequency [17], local derivatives [14, 10, 1] steerable filters [11], or invariant moments ([22]). In [23] these different descriptors have been compared, showing a best performance for SIFT-like descriptors.

The primitives we will be using in this work are local, multi-modal edge descriptors that were introduced in [20]. In contrast to the above mentioned features these primitives focus on giving a semantically and geometrically meaningful description of the local image patch. The importance of such a semantic grounding of features for a general purpose vision front-end, and the relevance of edge-like structures for this purposes were discussed in [6].

The primitives are extracted sparsely at locations in the image that are the most likely to contain edges. This likelihood is computed using the intrinsic dimensionality measure proposed in [19]. The sparseness is assured using a classical winner take all operation, insuring that the generative patches

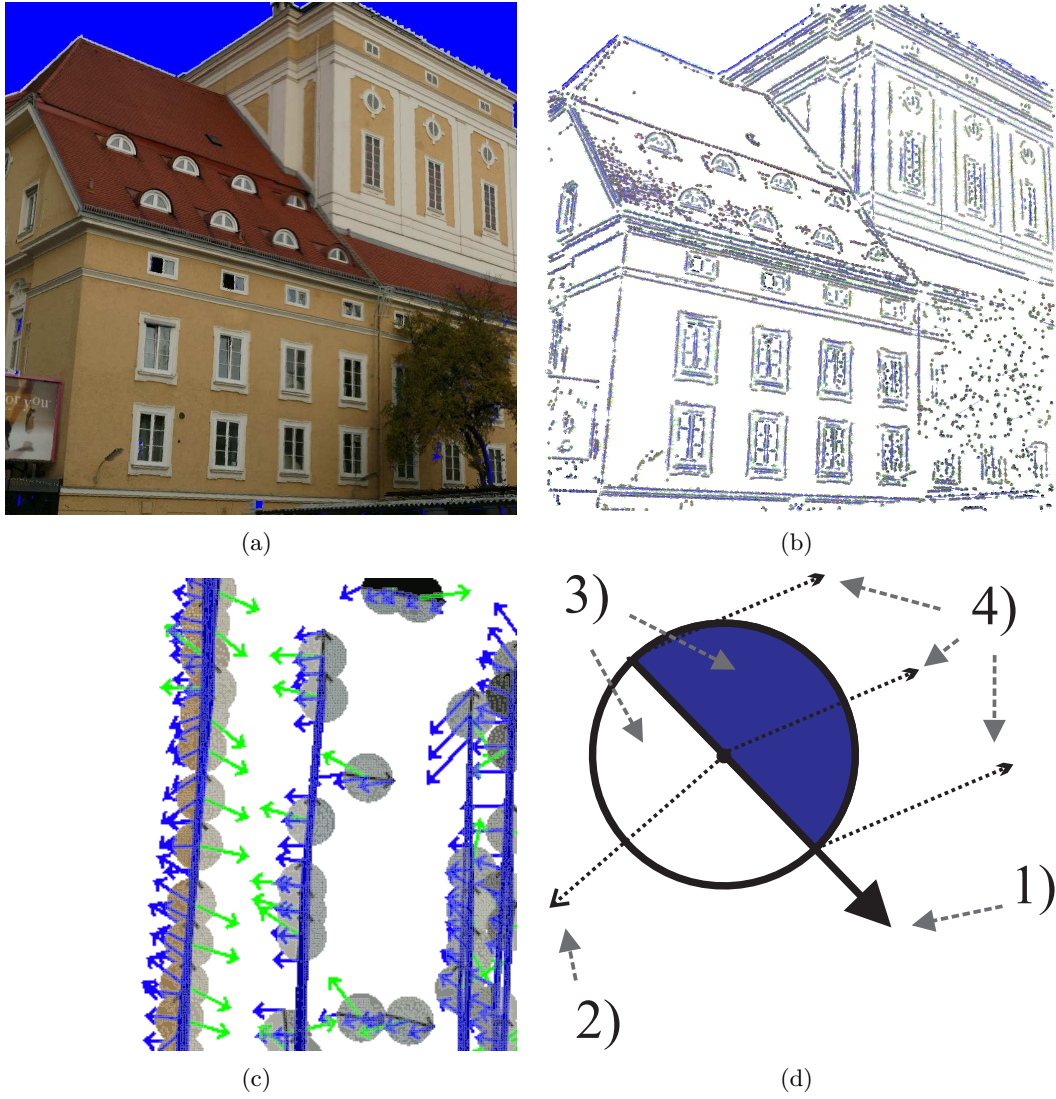


Figure 1: Illustration of the primitive extraction process from a video sequence. The figure shows in (a) one image from a video sequence on the right, then (b) the 2D-primitives extracted from this image, with a magnified version on (c). The blue lines between the primitives show the result of the perceptual grouping presented in [25] (d) describe the schematic representation of the 2D-primitives, where 1. shows the orientation of the primitive, 2. the phase, 3. the colour and 4. the optic flow.

of the primitives do not overlap (for details, see [21]). Each of the primitive encodes the image information contained by a local image patch. Multi-modal information is gathered from this image patch, including the position  $\mathbf{m}$  of the centre of the patch, the orientation  $\theta$  of the edge, the phase  $\omega$  of the signal at this point, the colour  $\mathbf{c}$  sampled over the image patch on both sides of the edge and the local optical flow  $\mathbf{f}$ . Consequently a local image patch is described by the following multi-modal vector:

$$\boldsymbol{\pi} = (\mathbf{m}, \theta, \omega, \mathbf{c}, \mathbf{f}, \rho)^T, \quad (1)$$

that we will name *2D primitive* in the following. In this equation  $\mathbf{m}$  refers to the position of the centre of the primitive in the image,  $\theta$  is the orientation of the primitive,  $\omega$  is the phase,  $\mathbf{c}$  is the colour value,  $\mathbf{f}$  is the local optical flow and  $\rho$  is the size of the primitive — see figure 1.

Note that these primitives are of lower dimensionality than, e.g., SIFT (10 vs. 128) and therefore suffer of a lesser distinctiveness. Nonetheless, as shown in [25] that they are distinctive enough for a reliable stereo matching if the epipolar geometry of the cameras is known. Furthermore, their semantic in terms of geometric and appearance based information allow for a good description of the scene content. It has been previously argued in [6] that edge pixels contain all important information

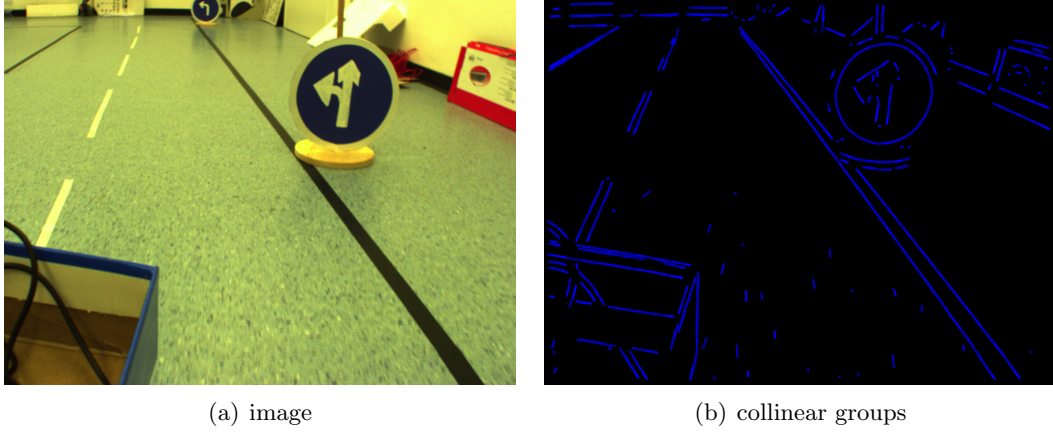


Figure 2: Collinear groups extracted from a sample image.

in an image. As a consequence, the ensemble of all primitives extracted from an image describe the shapes present in this image.

Advantageously, the rich information carried by the 2D-primitives can be reconstructed in 3D, providing a more complete scene representation. Having geometrical meaning for the primitive allows to describe the relation between proximate primitives in terms of perceptual grouping.

In a stereo scenario a 3D-primitive  $\Pi$  can be computed from two corresponding 2D-primitives (see figure 1 and [25]): such that we have a projection relation:

$$\mathcal{P} : \Pi \rightarrow \pi . \quad (2)$$

A 3D-primitive  $\pi$  is described by the vector:

$$\Pi = (M, \Theta, \Omega, C)^T, \quad (3)$$

where  $M$  is the location in space of the centre of the primitive,  $\Theta$  is its orientation vector,  $\Omega$  is its phase and  $C$  holds the colour on both sides of the primitive.

### 3 Relations between primitives

In [15] a variety of relations that can be drawn between visual primitives were reviewed. In this paper we will focus on the following:

#### 3.1 Collinearity

In [25] we proposed a simple scheme for grouping primitives that describe the same (smooth) contour of the scene. Herein we will assume that objects are delimited by piecewise smooth contours, joined by *junctions*. We will hereafter call *contour* these smooth sections.

Figure 2 shows the contours extracted by the grouping mechanism described in [25].

#### 3.2 Proximity

The proximity relation is the fact that the two primitives, when re-projected onto both views are distant of less than a certain radius. The likelihood for a random occurrence of this relation is:

$$p(d_E(a, b) < \tau_E) = \frac{(\tau_E)^2}{\rho^2} p(\pi) \quad (4)$$

where  $p(\pi)$  is the prior probability for the extraction of a primitive at a location.

$$p(\pi) = \frac{cr}{\#(\pi)\rho^2} \quad (5)$$

for a  $c \times r$  image where  $\#(\pi)$  primitives were extracted. Note that this two-dimensional definition of proximity is extended to 3D by enforcing that the re-projections on both image planes of the two 3D-primitives be proximate according to 2D definition.

### 3.3 Parallelism

We define the parallelism between two primitives as follows:

**Definition 1.** *Two primitives are said parallel if they share the same orientation.*

Therefore, collinearity is defined as follows:

$$|(a, b) = \text{acos}(\Theta_a \cdot \Theta_b) \quad (6)$$

If we consider that  $|(a, b)$  is always between  $[-\frac{\pi}{2}, +\frac{\pi}{2}]$  and if we consider as parallel all primitive pair  $(a, b)$  such that  $|(a, b) < \tau_{\text{coll}}$ , where  $\tau_{\text{coll}}$  is the tolerance of the parallelism definition, then we have:

$$p_{\text{prior}}(\text{coll}(a, b) < \tau_{\text{coll}}) = \frac{\pi}{\tau_{\text{coll}}} \quad (7)$$

assuming normal distribution.<sup>1</sup>

### 3.4 Coplanarity

Coplanarity was defined in [15]. Note that the shape of circular contours tend to be inaccurately reconstructed (due to the nearly horizontal parts of the curve). Therefore the coplanarity relation is not very robust on circular structures.

### 3.5 Co-colourity

We expect contours of the same surface to be co-colour. The co-colourity relation capture this prior knowledge about surfaces of the world. We will make use of this relation in conjunction with the parallelism and coplanarity relations to compensate for their relative statistical weakness. Co-colourity is fully described in [15].

## 4 Relations between contours

As stated before, the relations between two primitives, taken individually are still statistically weak events. Moreover, we argued in [25] that contours and not primitives (that are merely local descriptors sampled from scene contours) should be used for scene description.

Therefore we will extend the relations mentioned in the previous section onto contours. In extending the definition to collinear groups, we want to generate *rarer*, and therefore more salient, events.

### 4.1 Symbolic representation of contours

From the pairwise good continuation relation proposed in [25] we propose to extract the whole contour by using a classical transitivity relation.

**Definition 2.** *If primitive A and B are linked, and B and C are linked, then A, B and C are part of the same contour.*

We describe the resulting contours with the four following measures:

---

<sup>1</sup>Given that horizontal and vertical edges are more common in natural scenes than other orientations, this assumption of a normal distribution do not hold. Nevertheless it is good enough as a working hypothesis. Having a proper model of the orientation co-occurrence would only serve to weight less horizontal collinear segments than other orientations, which would not serve any purpose for SVE extraction.



### 4.3 Coplanar contours

Building onto the definition of coplanarity between two primitives, we define that a primitive  $a$  is coplanar to a contour  $B = (b_0, \dots, b_n)$  iff

$$\text{cop}(a, B) \text{ iff } \frac{\# \text{cop}(a, b_i)}{\# B} < r \quad (10)$$

where  $r$  is a ratio, that we set to  $r = 0.8$  for our experiments. A higher value will lead to a stricter definition whereas a lower one will find more cases of coplanarity.

Then we define the coplanarity between two contours  $A = (a_0, \dots, a_n)$  and  $B = (b_0, \dots, b_n)$  as

$$\text{cop}(A, B) \text{ iff } \begin{cases} \frac{\# \text{cop}(a_i \in A, B)}{\# A} < r \\ \frac{\# \text{cop}(b_j \in B, A)}{\# B} < r \\ \min_{a_i \in A, b_j \in B} (d_E(a_i, b_j)) < \tau_E \end{cases} \quad (11)$$

In other words, two groups are coplanar if a sufficient ratio of the primitives thereof are coplanar. Note that this can only occur for groups with a strong planarity. This is illustrated in figure 3(c), where the dashed circle shows the proximity criterion, and the dashed lines represent the two other criteria.

## 5 Results and discussion

We applied these relations to some video simple sequences featuring some sample objects. For the purpose of these experiments we proceeded in extracting the SVEs in two steps:

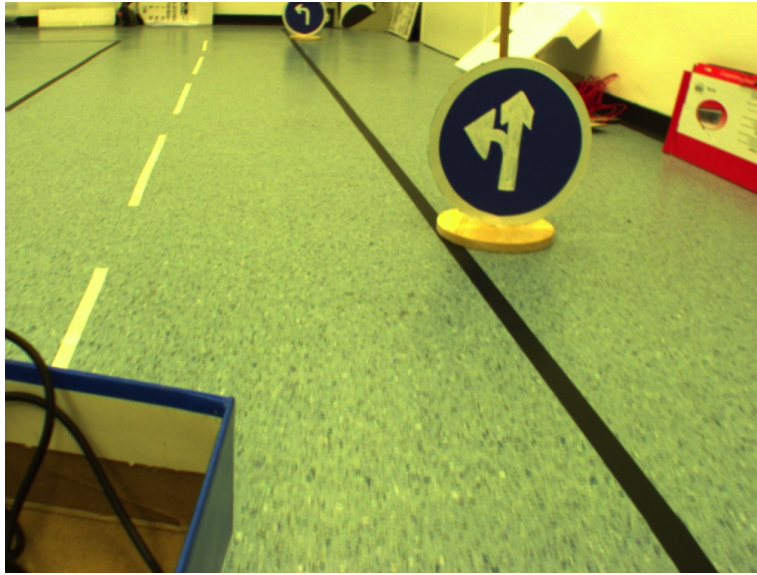
1. extract 3D contours, as in [15].
2. Compute the relations between all contours.
3. merge all linked contour into one SVE.

Note that this method is only used for experimentation purpose. In the future it would be preferable to keep the relational structure between all primitives instead of merging them all into one group. We applied this method for two combinations of relations:

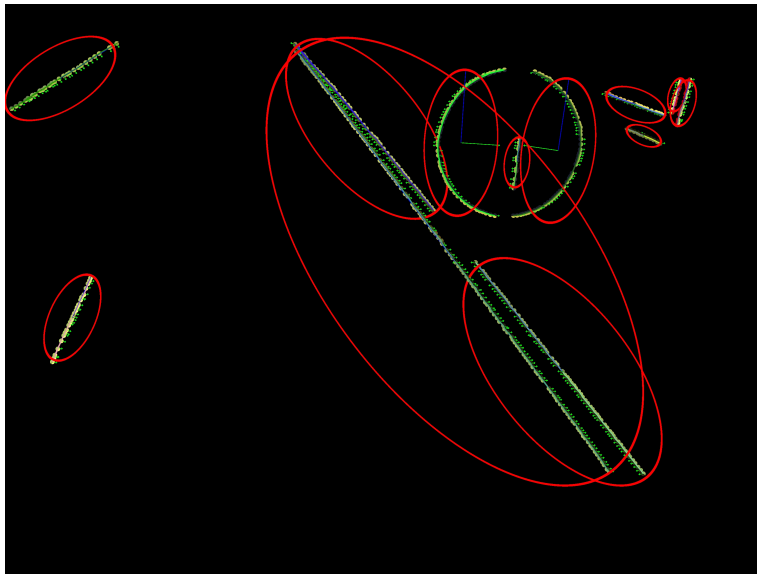
**Parallelism + co-colourity:** In this case we only considered the relation of parallelism. We also required that two primitives be co-colour in order to be considered as parallel. The results of this method applied to a driving scene are shown in figure 4. There we can see that the different parts of the white line are merged together. On the other hand, the two crescent-shaped parts of the traffic sign are left separate. Also the three lines on the ground, although parallel are not merged. This is due to the proximity constraint that we enforced in the definition of contour parallelism.

**Coplanarity + co-colority** For the second experiment we replaced the parallelism relation by the somewhat weaker coplanarity relation. Here again we required that the co-colourity be respected to consider two primitives as coplanar. The results applied to a sequence showing a traffic sign are shown in figure 5. Note that both sides of the support and both sides of the traffic signs a successfully grouped. The horizontal part of the traffic sign suffer from a reconstruction of low accuracy (because it is horizontal) and therefore the coplanarity is too weak to merge it. The results when applied to another scene featuring two mugs on a table are shown in 6. There we can see that the corners of the table are successfully merged. Here again the horizontal parts lead to problems: the horizontal border of the table is not grouped. Moreover because the reconstruction of the circular opening of the cups is inaccurate due to the horizontal (and curved) parts, some parts of the cups are found coplanar where they should not.

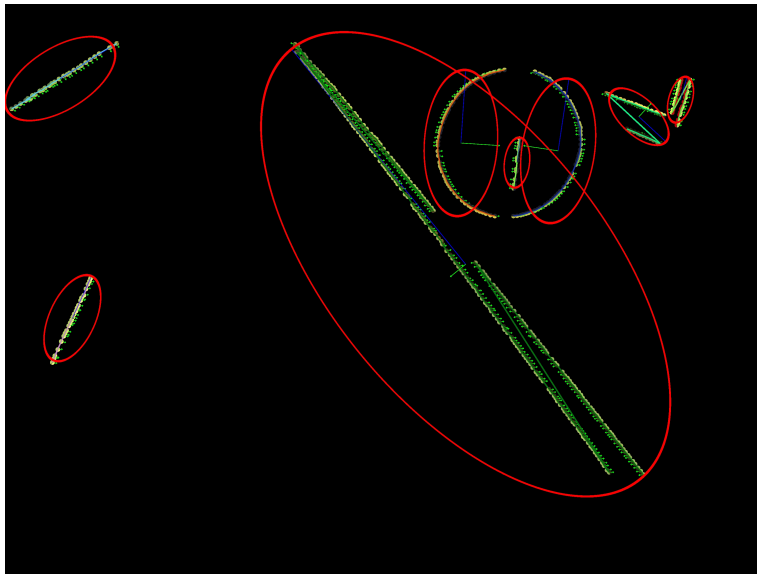




(a)



(b)



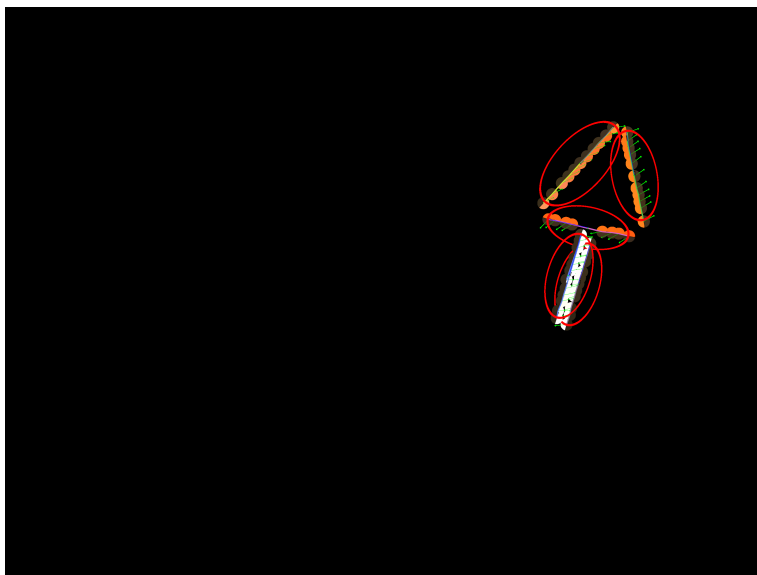
(c)

Figure 4: Example of the extraction of Visual Gestalts using good continuation (b) and parallelism + co-colourity (c) (the red ellipses show the Gestalts).

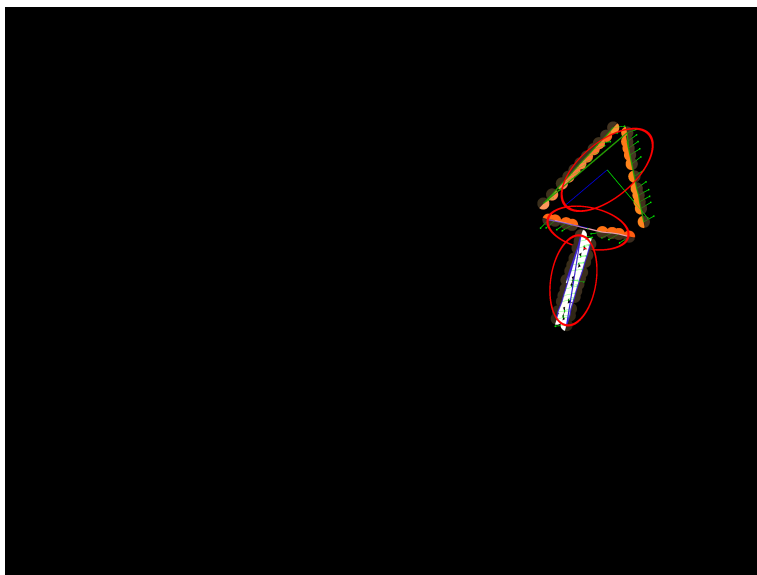




(a)

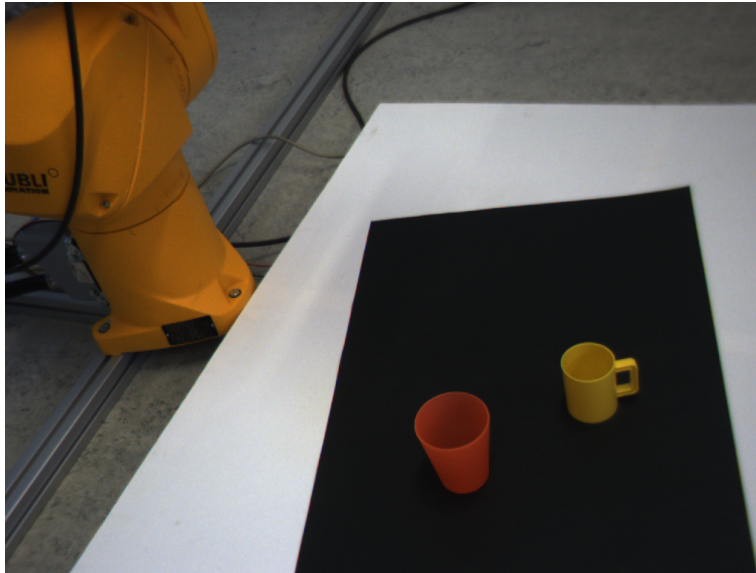


(b)

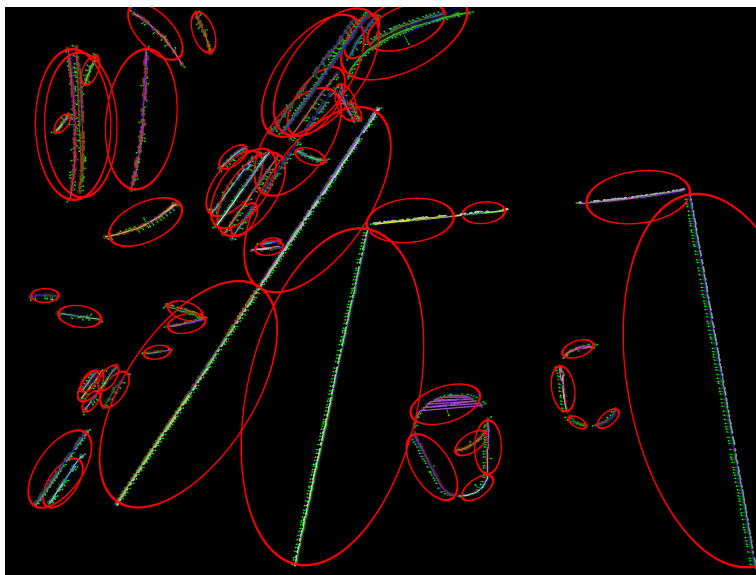


(c)

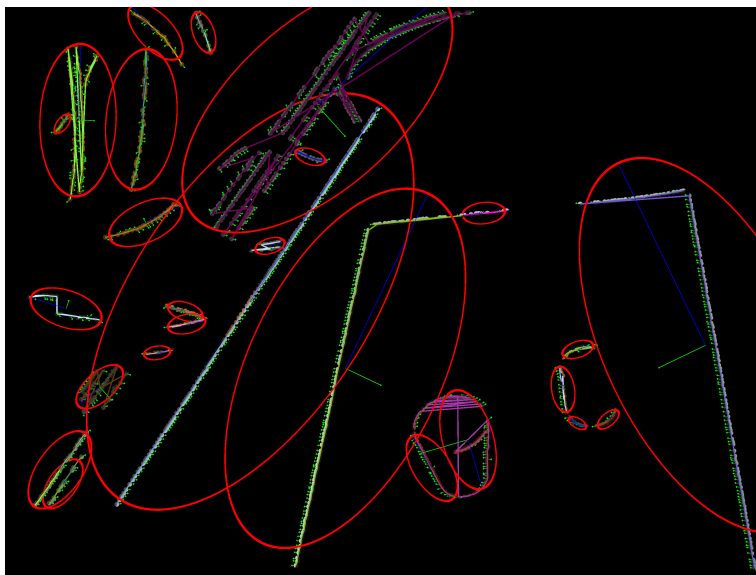
Figure 5: Example of the extraction of Visual Gestalts using good continuation (b) and coplanarity + co-colourity (c) (the red ellipses show the Gestalts).



(a)



(b)



(c)

Figure 6: Example of the extraction of Visual Gestalts using good continuation (b) and coplanarity + co-colourity (c) (the red ellipses show the Gestalts).

These results show that relations, when extended to collinear groups becomes stronger predictor of object structure than when applied to basic primitives. Although the group relations are directly based on the primitives' relations defined in [15], this extension offer a considerably lower likelihood of accidental occurrence.

From these preliminary results, we propose to design a hierarchical architecture for representing explicitly complex structures in the scene and evaluating the saliency thereof.

**3D contours extraction:** using the process explained in [15], we propose to extract contours from the image representation provided by the primitives.

**Evaluation of inter-contour relations** in our case we will limit to 1) parallelism + co-colority; and 2) coplanarity + co-colority. Future work should focus on integrating symmetry and the relations provided by the addition of junction primitives (see [16]) into the scheme.

**Design good structure to represent shapes** As a result of the above-mentioned mechanism, strongly structured objects should appear as densely linked in the resulting graph. If we consider the simple case of a coloured square, we would have each side of the square as a contour. Opposed sides would be parallel and contiguous sides would be coplanar. The advantage of a shape representation based on 3D-contours is that it is largely independent from viewpoint, scaling and sampling. For example, [26] proposed to use a similar hierarchical shape representation for the purpose of object recognition.

**Feedback the information to lower level processes** E.g. the stereopsis. A stereopsis of good quality is essential for the grouping process to perform well. On the other hand, at each level of the grouping hierarchy new information is obtained that could be used to disambiguate stereopsis, in a similar way that the lowest level grouping information was used in [24]. Chung and Nevatia [2] used a similar approach to stereo disambiguation with the notable difference that they restricted themselves to monocular grouping. We argue here that perceptual grouping and 3D-reconstruction should be processed in parallel using extensive communication between the two processes. Our objective is to address these points in the upcoming year, in order to obtain a higher level symbolic scene representation.

## References

- [1] A. Baumberg. Reliable Feature Matching across Widely Separated Views. In *Proc. Conf. Computer Vision and Patter Recognition*, pages 774–781, 2000.
- [2] R. Chung and R. Nevatia. Use of monocular groupings and occlusion analysis in a hierarchical stereo system. *Computer Vision and Image Understanding*, 62(3):245–268, 1995.
- [3] Cordelia Schmid and Roger Mohr and Christian Baukhage. Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.
- [4] D. Crevier. A probabilistic method for extracting chains of collinear segments. *Computer Vision and Image Understanding*, 76(1):36–53, october 1999.
- [5] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.
- [6] J. H. Elder. Are edges incomplete ? *International Journal of Computer Vision*, 34:97–122, 1999.
- [7] J. H. Elder, A. Krupnik, and L. A. Johnstone. Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):661–674, 2003.
- [8] J. H. Elder and S. H. Zucker. Evidence for boundary specific grouping. *Vision Research*, 38(1):143–152, 1998.
- [9] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric ViewPoint*. MIT Press, 1993.

- [10] Frederik Schaffalitzky and Andrew Zisserman. Multi-view Matching for Unordered Image Sets, or “How Do I Organize My Holiday Snaps?”. *Lecture Notes in Computer Science*, 2350:414–431, 2002. in Proceedings of the BMVC02.
- [11] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE-PAMI*, 13(9):891–906, 1991.
- [12] W. Geisler, J. Perry, B. Super, and D. Gallogly. Edge co-occurrence in natural images predicts contour grouping performance. *Vision Research*, 41:711–724, 2001.
- [13] C. G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.
- [14] J. J. Koenderink and A. J. van Doorn. Representation of Local Geometry in the Visual System. *Biological Cybernetics*, 55:367–375, 1987.
- [15] S. Kalkan, N. Pugeault, and N. Krüger. Perceptual operations and relations between 2d or 3d visual entities. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-3, 2007.
- [16] S. Kalkan, S. Yan, F. Pilz, and N. Krüger. Improving junction detection by semantic interpretation. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.
- [17] P. Kovessi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1(3):1–26, 1999.
- [18] N. Krüger. Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129, 1998.
- [19] N. Krüger and M. Felsberg. A continuous formulation of intrinsic dimension. *Proceedings of the British Machine Vision Conference*, pages 261–270, 2003.
- [20] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *Interdisciplinary Journal of Artificial Intelligence & the Simulation of Behaviour, AISB Journal*, 1(5):417–427, 2004.
- [21] N. Krüger, N. Pugeault, and F. Wörgötter. Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-4, 2007.
- [22] Luc Van Gool and Theo Moons and Dorin Ungureanu. Affine / Photometric Invariants for Planar Intensity Patterns. *Lecture Notes In Computer Science*, 1064:642–651, 1996. in Proceedings of the 4th European Conference on Computer Vision — Volume 1.
- [23] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
- [24] N. Pugeault, F. Wörgötter, , and N. Krüger. Structural Visual Events. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-1, 2007.
- [25] N. Pugeault, F. Wörgötter, and N. Krüger. Multi-modal scene reconstruction using perceptual grouping constraints. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR’06)*, 2006.
- [26] A. Selinger and R. C. Nelson. A perceptual grouping hierarchy for appearance based 3d object recognition. *Computer Vision and Image Understanding*, 76(1):83–92, october 1999.

## **Appendix L**

### ***Stereo Reconstruction Uncertainty and Reliability 3D-Feature***

N. Pugeault, S. Kalkan, E. Baseski, F. Woergoetter and N. Krueger

*( to be submitted )*

# Reconstruction Uncertainty and 3D Relations

Nicolas Pugeault

University of Edinburgh, United Kingdom

`npugeaul@inf.ed.ac.uk`

Sinan Kalkan

University of Göttingen, Germany `sinan@chaos.gwdg.de`

Emre Baseski

Syddansk University, Denmark `emre@mmi.sdu.dk`

Florentin Woërgöetter

University of Göttingen, Germany `worgott@bccn-goettingen.de`

Norbert Krüger

Syddansk University, Denmark `norbert@mmi.sdu.dk`

August 31, 2007

## Abstract

In this paper, we first propose an analytic formulation for the position's and orientation's uncertainty of local 3D line descriptors reconstructed by stereo. We evaluate these predicted uncertainties with Monte Carlo simulations, and study their dependency on different parameters (position and orientation). In a second part, we use this definition to derive a new formulation for inter-features distance and coplanarity. These new formulations take into account the predicted uncertainty, allowing for better robustness. We demonstrate the positive effect of the modified definitions on some simple scenarios.

## 1 Introduction

Many computer vision applications make use of 3D objects models, provided to the system. Because these models are designed specifically for the task at hand, they can be precise, rich, and concise at

the same time, and thereby simplify greatly reasoning problems. A common problem then is to relate the visually reconstructed 3D information about the scene with this accurate model knowledge. Local descriptors, as presented in section 2, have the advantage of being numerous and of describing the shape of the objects being witnessed. Their downside is that they describe only a small part of the object, and therefore are not very distinctive, and that objects are not uniquely described by local descriptors, due to sampling. Therefore it is advantageous to consider, beside the primitives themselves, relations between them: distance, collinearity, coplanarity, etc. For example, a square is described by parallel and orthogonal strings of collinear 3D-primitives, positioned at fixed distance one from the other — see [1] for a discussion of visual representation with primitives' relations.

When using exogenous knowledge about the objects in the scene, and the relations that define them, one needs to consider the fact that primitives are only reconstructed up to a certain precision. Thus, inter-

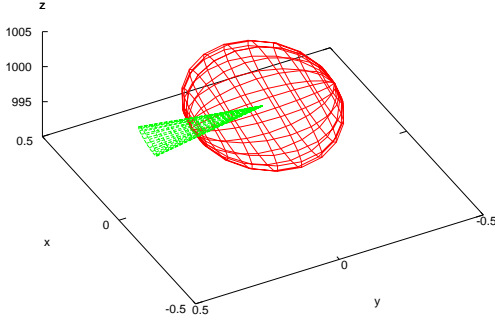


Figure 1: Illustration of the uncertainty. The red ovoid shows the position’s uncertainty, and the green cone the orientation’s uncertainty. The axes of the ellipse and the cone are computed from the Eigen-values and associated Eigen-vectors of the covariance matrices.

primitives relations can only be defined up to a certain tolerance that depends on primitive uncertainty. Moreover, the *selectivity* of a relation is inversely proportional to this tolerance. A primitive’s uncertainty is function of image noise, calibration imprecision, and inaccuracies in primitive extraction, stereopsis, and reconstruction processes. This leads to large variations in primitives’ uncertainty across the visual field. Position’s uncertainty can be pictured as an ovoid in space, containing the possible reconstructed positions; similarly, orientation’s uncertainty forms a distorted cone. This is illustrated in Fig. 1. In this work we will model parameters uncertainty by their covariance matrices, and predict their propagation using a analytical first order approximation proposed by [4, 5, 2]. This is discussed in the first part of this paper, in section 3.

The computation of inter-primitives relations can be severely affected by the imprecision in the 3D-primitives’ reconstruction. For example, consider the collinearity relation. If we make abstraction of the primitives’ imprecision, we can use the standard mathematical definition: two 3D-primitives are collinear if their orientation is parallel to the line

that joins them. Now if we add some imprecision in the reconstruction process, these orientations will be slightly different. Normally this could be addressed by setting a threshold on the orientation difference, but the primitives’ uncertainty depends on parameters such as its orientation and position in space. In other words, there is no single threshold that can be set to define collinearity adequately for all cases. In the second part of this paper, in section 4, we will consider two relations: distance 4.1 and coplanarity 4.2. For each relation we propose a classic Euclidian formulation, and a second one taking into account the primitives’ uncertainty, in a manner reminiscent of the Mahalanobis distance. We compare the robustness (how regularly correct primitives pairs are identified) and selectiveness (how often primitives are erroneously paired) of the two formulations.

## 2 The primitive-based vision framework

In this paper we make use of a framework proposed in [8]. This representation describes the image in terms of a sparse set of local, multi-modal line descriptors called *2D-primitives*. In this work we are only interested in the primitives’ position ( $\mathbf{m}$ ) and local orientation (defined by the tangent vector  $\mathbf{t}$ ).<sup>1</sup>

A stereo-pair of 2D-primitives allows to reconstruct a *3D-primitive*: a local 3D contour descriptor (which position is defined by  $\mathbf{M}$  and orientation by the tangent vector  $\mathbf{T}$ ). Fig. 2 illustrates the 2D-primitive extraction and 3D-primitive reconstruction processes: (a) shows an image from an indoor navigation scenario; (b) shows the extracted 2D-primitives, with a detail on the traffic sign in (c); finally, (d) shows the 3D-primitives reconstructed by stereo.

## 3 Computing uncertainties

The computation of visual processes’ uncertainty has been studied by several groups. Verri and Torre [13]

<sup>1</sup> Primitives also hold some aspect parameters such as colour and phase, that are useful for, e.g., the stereo-matching process. See [8].

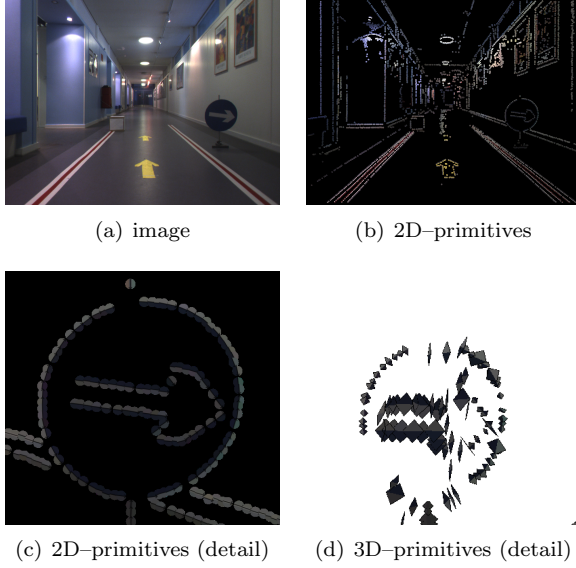


Figure 2: Illustration of the primitive-based vision framework presented in [8] and used in this study.

studied reconstructed points' depth accuracy, and found that the length of the baseline is critical for the accuracy. Rodriguez and Aggarwal [11] proposed to approximate reconstruction uncertainty by the *relative range error*, and Mandelbaum and colleagues [9] handle the depth uncertainty as a minimax risk confidence interval. These works only consider the depth uncertainty in the case of point reconstruction. The proposed formulations do not allow for an easy inclusion of additional parameters. Hartley and Zisserman [6] argue that the angle between the optical rays back-projected by a pair of image points yields a better estimate of the reconstructed point's covariance than the disparity. Wolff [14] discussed the stereo-reconstruction of lines, and propose an estimation of the reconstructed orientation's uncertainty, demonstrating that reconstructing lines as an intersection of planes lead to a better accuracy than reconstructing the lines' endpoints. The proposed analytical derivation is less general specific than the one used in this paper. Clarke [2] also suggests to use Monte-Carlo simulation to estimate uncertainty, but points out the extreme computational cost of this approach.

We argue that this approach is unpractical when taking additional parameters into account (orientation, sparseness, cameras' projection matrices), but provides an efficient way to evaluate an analytic derivation (see section 3.3). Heuel and colleagues [7] proposed a 3D line reconstruction using uncertain geometry. Their approach focuses on polyhedral objects, whereas the primitive-based framework used herein allows the representation of curved contours using local edge descriptors. This locality aspect requires us to reconstruct a position on the reconstructed 3D-line.

In this work, we first estimate the 2D-primitive's extraction process uncertainty, then describe how it propagates to 3D-primitives, using the formulation proposed by [4, 5, 2]. Additional uncertainties stem from the projection matrices (these should be obtained from camera calibration), from stereo matching (an estimation is proposed here), and local curvature (that we will neglect in this paper). We model parameters' uncertainties with their covariance matrices (see, e.g., [2]).

**Uncertainty propagation** Given a function  $y = f(\mathbf{m})$ , where  $\mathbf{x}$  and  $\mathbf{y}$  are vectors with associated covariance matrices  $\mathbf{\Lambda}_{\mathbf{x}}$  and  $\mathbf{\Lambda}_{\mathbf{y}}$ , a first order Taylor series expansion gives us:

$$f(\mathbf{x} + \Delta\mathbf{x}) = f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \Delta\mathbf{x} + O(\|\Delta\mathbf{x}\|^2) \quad (1)$$

from there [2] derives that the relation between the covariance matrices of  $\mathbf{m}$  and  $\mathbf{y}$  is approximated by the relation

$$\mathbf{\Lambda}_{\mathbf{y}} \approx \nabla f \cdot \mathbf{\Lambda}_{\mathbf{x}} \cdot \nabla f^\top \quad (2)$$

where  $\nabla f$  is the Jacobian matrix for the function  $f$ . This is the main result used hereafter to estimate uncertainties' propagation during stereo reconstruction. In the following we will equivalently denote  $\Lambda = \sigma^2$  the variances of scalar values, and  $\mathbf{\Lambda}$  the covariance matrices of vector quantities. Also, in the one-dimensional case,  $\nabla f(x) = \frac{\partial f(x)}{\partial x}$  is the derivative of  $f(x)$ .



### 3.1 2D-primitive uncertainty

Consider a set of 2D-primitives  $\pi_i$ , we want to estimate the 2D-primitives' position and orientation uncertainty. In [8], the 2D-primitives' position mean square error was evaluated to  $\epsilon < 0.25^2$  pixels. For simplicity, we assume an isotropic error in the following<sup>2</sup> and approximate the primitive's localisation covariance matrix  $\Lambda_{\tilde{\mathbf{m}}}$  by:

$$\Lambda_{\mathbf{m}} \simeq \begin{pmatrix} \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (3)$$

Note that this covariance matrix corresponds to the of the 2D-primitive's homogeneous position  $\tilde{\mathbf{m}}$ , and therefore its third dimension's variance is null.

A 2D-primitive's orientation variance is approximated to its mean square error, evaluated in [8] to  $\Lambda_{\theta} \simeq 0.03^2$  radians.

### 3.2 Reconstruction uncertainty

We then study the propagation of 2D-primitives' uncertainty during stereo-reconstruction, and estimate the resulting 3D-primitives' uncertainty.

The relation between points in space and their projection in the image is defined by the camera's projection matrix  $\tilde{P}$  (see [5, 6]). In the following, and for the sake of simplicity, we assume that the cameras' parameters are known, and their projection matrix exact  $\Lambda_{\tilde{\mathbf{P}}} = \mathbf{0}_{12 \times 12}$ . In the general case, the projection matrix will be estimated empirically through a process called *calibration* that provides its uncertainty as a by-product [3].

Classical stereo-reconstruction tries to intersect two optical rays containing the possible origins of (or *back-projected* by) two corresponding points in two images. Because of imprecision, it is unlikely that the two lines intersect, and therefore the closest point to both rays is usually chosen. This approach is inadapted in the case of local line descriptors because the aperture problem makes reliable point matching impossible. On the other hand, [14] discussed that

accurate line matching could be achieved by intersecting the two planes back-projected from the lines in each image. Moreover, because primitives are *local* line descriptors we need a location along this line. This is obtained by intersecting the *line* containing the left 2D-primitive's position possible origins with the *plane* containing the right 2D-primitive's possible origins. The computation of the 3D-primitives' uncertainty is similar to the work of [2, 7], and is described in Appendices A and B.

### 3.3 Evaluation

We evaluate the quality of the uncertainties predicted by the above formulae, using a Monte Carlo simulation in a simple scenario. The focal length is set to  $f = 10^3$  and the baseline to  $b = 100$ , so that the optical centres of the cameras are located at  $\mathbf{C}_1 = (0, 0, 0)^\top$  and  $\mathbf{C}_2 = (b, 0, 0)^\top$ .<sup>3</sup>

Consider a 3D-primitive at a location  $\hat{\mathbf{M}} = (0, 0, 100)^\top$  and with an orientation  $\hat{\mathbf{T}}$ , projected on both image planes as  $\hat{\pi}^l$  and  $\hat{\pi}^r$ . Using a Monte Carlo simulation of 100.000 particles, we measured a relative error between predicted and measured covariance matrices  $\xi = \frac{\|\Lambda' - \Lambda\|}{\|\Lambda'\|}$  of  $\sim 3\%$  for position, and  $\sim 4\%$  for orientation.

We then investigated how the 3D-primitive's position and orientation impact the uncertainty thereof. We compared the trace  $tr(\Lambda)$  of the reconstructed position's covariance matrix (sum of the Eigen-values), at different locations in space (Figs. 3(a), 3(b), and 3(c) for different values of the  $x$  (horizontal),  $y$  (vertical), and  $z$  (depth) coordinates) and for different pairs of 2D-orientations (Fig. 4(a)).

These figures show that the reconstructed position's covariance is affected by the distance from the primitive to the cameras' optical centres and by the right 2D-primitive's orientation.

The trace  $tr(\Lambda_{\mathbf{m}})$  in Fig. 4(a) is mostly affected by  $\theta_2$ . This is due to the line reconstruction formula used in this work — see section 3.2. In this formulation, the right 2D-primitive's orientation is used

<sup>2</sup> Although the actual localisation error of 2D-primitives is more significant in the direction orthogonal to the contour's orientation.

<sup>3</sup> These values were chosen for simplicity, but are nevertheless plausible: they are similar to the calibration parameters of an actual stereo camera system.

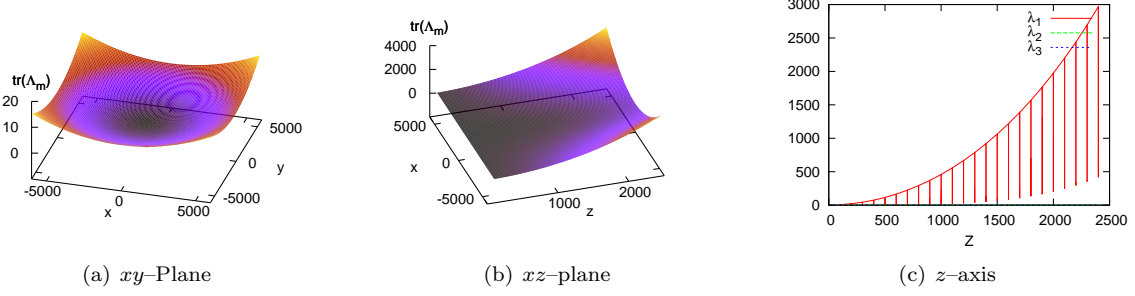


Figure 3: Traces of the covariance matrix  $\Lambda_M$ , for (a) different locations  $M = (x, y, 100)^\top$  on the  $xy$ -plane; (b) different locations  $M = (x, 0, z)^\top$  on the  $xz$ -plane; and (c) just considering the  $z$ -axis.

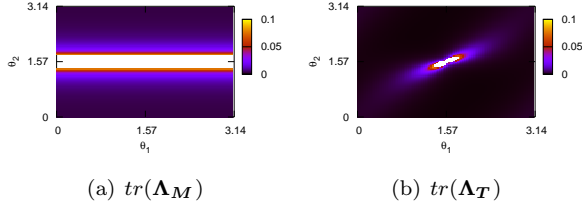


Figure 4: Effect of 2D-primitives' orientations on (a) the trace of  $\Lambda_M$ ; and (b) the trace of  $\Lambda_T$ .

to resolve the ambiguity that stems from the aperture problem (we compute the intersection between a back-projected *left* ray and a back-projected *right* plane, see Eq. (26)). This becomes impossible when the primitive's orientation is the same that the epipolar line's (in this case if  $\theta_2 = \frac{\pi}{2}$ ), and therefore the reconstructed 3D-primitive's position uncertainty increases to infinity for orientations close to  $\frac{\pi}{2}$ .

We then evaluated the 2D-primitives' orientation impact on the reconstructed 3D-primitive's orientation uncertainty. Fig. 4 plots the trace of the reconstructed orientation's covariance matrix for a point located at  $m = (0, 0, 100)^\top$ , reconstructed from different 2D-primitives' orientations. In this figure we see that the reconstructed orientation uncertainty increases when either of the 2D-primitive's orientation becomes close to  $\frac{\pi}{2}$ . When both orientations become close to  $\theta_1 = \theta_2 = \frac{\pi}{2}$  two primitives back-project the

same plane  $\mathcal{P}_1 = \mathcal{P}_2$ , and therefore their intersection is undefined.

## 4 Design of relations between 3D-primitives

In this section we consider distance and coplanarity between 3D-primitives, and propose definitions that take the uncertainties thereof into account, based on the Mahalanobis distance.

### 4.1 Distance between 3D-primitives

The first relation that we consider is the *distance* between two reconstructed 3D-primitives.

Consider the following scenario: three primitives, located at points  $A = (100, 100, z)^\top$ ,  $B = A + u$  ( $u = (a, 0, 0)^\top$ ) and  $C = (a + b + 100, 100, z)^\top$ , with a vertical orientation ( $a = 50$  and  $b = 10$ ). These points' projections on both image planes are subjected to a zero-mean Gaussian perturbation applied to the projected 2D-primitives' position and orientation, with a standard deviation of  $\sigma = 0.25$  and  $\sigma = 0.03$  respectively. Then we reconstruct the 3D-primitives  $\Pi_i$  as described in section 3.2. We assume that we have the model knowledge that  $B$  lies at a distance of exactly  $b$  from  $A$ , and we want to use this knowledge to identify the primitives  $\Pi$  that belong to  $A$ ,  $B$ , and  $C$ . This is illustrated in a concrete scenario in Fig. 5. In this scenario, we know that the

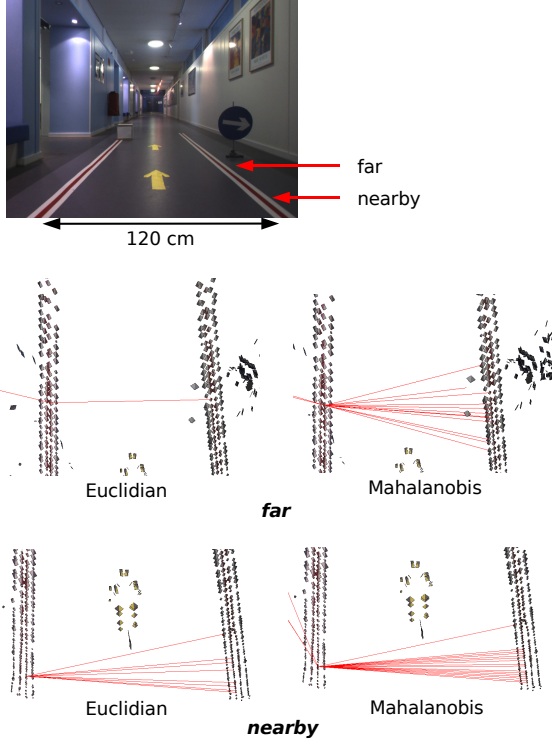


Figure 5: Example of the distance relation, for nearby and far primitives pairs. The red lines indicate valid pairs.

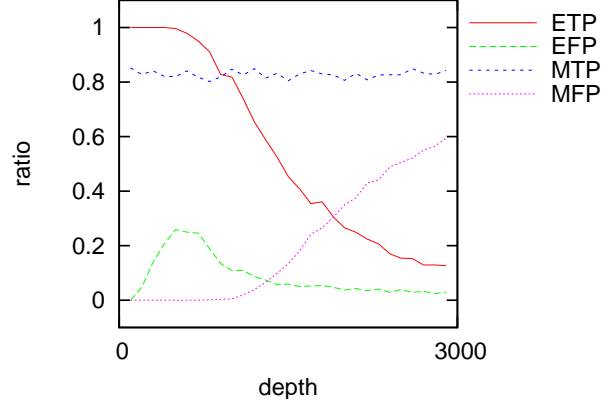


Figure 6: Comparison of the robustness of Euclidian (E) and Mahalanobis (M) distances, for the values  $a = 50$ ,  $b = 5$ ,  $\alpha = 20$ , and  $\beta = 5$ .

two red lines on the ground, delimiting the road, are separated by a distance  $a$  of 120cm. Using this world knowledge, we search for pairs of primitives that are separated by this distance, plus or minus 10cm. The figure shows the valid pairs for nearby and far 3D-primitives. In each case the red lines indicate with which other primitive it forms a valid pair according to each definition for distance.

We compare the performance of different distance measures for this task:

**Euclidian distance threshold (E):** We defined the threshold on the Euclidian distance as follows:

$$|d_E^2(\Pi_1, \Pi_2) - \|y\|| < \alpha^2 \quad (4)$$

where  $d_E^2(\Pi_1, \Pi_2)$  stands for the squared Euclidian distance between  $\Pi_1$  and  $\Pi_2$ .

**Mahalanobis distance (M):** The second criterion is based on the Mahalanobis distance:

$$((\Pi_1 + u) - \Pi_2)^\top (\Lambda_{\Pi_1} + \Lambda_{\Pi_2})^{-1} ((\Pi_1 + u) - \Pi_2) < \beta \quad (5)$$

#### 4.1.1 Evaluation

We compared the performance and robustness of both formulations using artificial images. We set  $a = 50$ ,  $b = 5$ ,  $\alpha = 20$ , and  $\beta = 5$ . The results are summarised in Fig. 6, the true positives curves (ETP and MTP) express the ratios of experiments wherein the reconstructed 3D-primitives  $\mathbf{A}'$  and  $\mathbf{B}'$  comply with the criterion (respectively E and M). The false positive curves (EFP and MFP) express the ratios of experiments wherein the reconstructed 3D-primitives  $\mathbf{A}'$  and  $\mathbf{C}'$  satisfy the criterion. In this figure, we see that the number of true positive of the Euclidian criterion (ETP) decreases with depth.<sup>4</sup> On the other hand, the ratio of true positive (MTP) is stable for the Mahalanobis distance. The false positives (MFP) increase progressively for large uncertainties, when the distribution of  $\mathbf{B}$  and  $\mathbf{C}$  overlap significantly. This shows that the normalised Mahalanobis distance is better suited for drawing spatial relations between reconstructed 3D-primitives.

This trend is illustrated qualitatively on real images in figure 5. There we have the values:  $a = 120$ ,  $\alpha = 10$ , and  $\beta = 0.5$ .

#### 4.2 Coplanarity relation

The second relation we studied is the coplanarity between two reconstructed 3D-primitives. As before, we consider three 3D-primitives,  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$ , with  $\text{cop}(\mathbf{A}, \mathbf{B}) = 1$  and  $\text{cop}(\mathbf{A}, \mathbf{C}) \simeq 0.70$  — this means an angle of  $\frac{\pi}{4}$ . The 3D-primitives are projected onto the image planes as before, the same Gaussian perturbation is applied, and both coplanarity criteria are applied to the reconstructed 3D-primitives  $\Pi_i$ .

Coplanarity is defined as follows:

$$\text{cop}(\Pi_1, \Pi_2) = (\mathbf{V} \times \mathbf{T}_1) \cdot (\mathbf{V} \times \mathbf{T}_2) \quad (6)$$

where  $\mathbf{V} = \frac{1}{\|\mathbf{M}_2 - \mathbf{M}_1\|} \cdot (\mathbf{M}_2 - \mathbf{M}_1)$ . By using Eq.(13) in Eq.(6) we obtain the variance of the coplanarity

<sup>4</sup> Note that the performance of the Euclidian distance (E) could be improved for a certain region of the space by altering  $\alpha$ . Nonetheless, the general trend will be the same: larger  $\alpha$  lead to more false positives for nearby structures, and the number of true positives tend to zero for far structures.

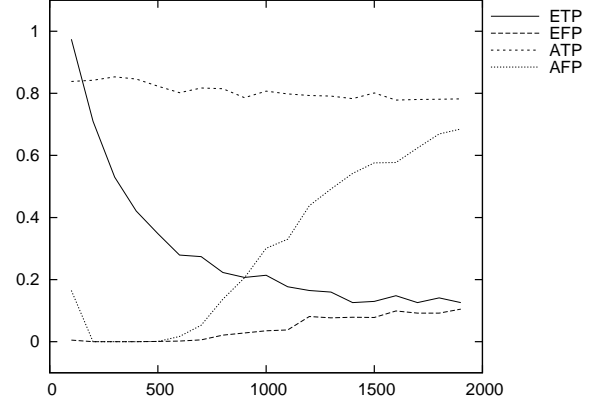


Figure 7: Proportion of coplanar pairs correctly labelled, using a fixed (E) and a variance dependent threshold (A), respectively.

measure:

$$\Lambda_{\text{cop}} = \begin{pmatrix} \eta_2^\top & \eta_1^\top \end{pmatrix} \cdot \begin{pmatrix} \Lambda_{\mathbf{V} \times \mathbf{T}_1} & \\ & \Lambda_{\mathbf{V} \times \mathbf{T}_2} \end{pmatrix} \cdot \begin{pmatrix} \eta_2 \\ \eta_1 \end{pmatrix} \quad (7)$$

with  $\eta_i = \mathbf{V} \times \mathbf{T}_i$  the normal to the plane formed by the orientation  $\mathbf{T}_i$  and the points  $\mathbf{M}_1$  and  $\mathbf{M}_2$ . Therefore, we propose the two following criteria for coplanarity:

**Euclidian coplanarity:** The first definition simply applies a threshold on the coplanarity value:

$$1 - \text{cop}(\Pi_1, \Pi_2) < \alpha \quad (8)$$

**Mahalanobis coplanarity:** The second definition makes use of the estimated coplanarity variance to derive a Mahalanobis-like criterion:

$$\Lambda_{\text{cop}} \cdot (1 - \text{cop}(\Pi_1, \Pi_2))^2 < \beta \quad (9)$$

These two criteria, in Eq. 8 and 9, are compared in Fig 7, for values  $\alpha = 0.01$  and  $\beta = 0.5$ . In this figure: ETP is the ratio of cases where Eq. 8 is verified between  $\mathbf{A}'$  and  $\mathbf{B}'$ , EFP where it is between  $\mathbf{A}'$  and  $\mathbf{C}'$ ; ATP the ratio where Eq. (9) is satisfied between  $\mathbf{A}'$  and  $\mathbf{B}'$  and AFP the ratio where it is satisfied between  $\mathbf{A}'$  and  $\mathbf{C}'$ . In Fig. 7 we see that the ratio

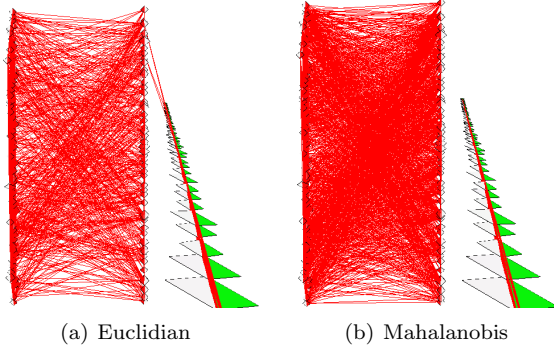


Figure 8: Illustration of the coplanar pairs extracted. The red lines show the primitives coplanar near (bottom) and far (top) from the camera.

ETP reduces quickly with the increase of depth. The ATP ration, on the other hand, is stable, while the AFP ratio increases with depth. This shows that the variance adapted threshold is a more robust criterion for reconstructed features’ coplanarity than the naive Euclidian criterion, and this across a wide range of depth.

The result is further illustrated in Fig. 8. We see that when using the Mahalanobis version, the coplanar structures (red) are more densely connected than when using the Euclidian threshold, thus coplanarity is more reliably asserted. Furthermore, it is visible that the Euclidian criterion interpretes some of the farther green primitives as coplanar with the red ones.

## 5 Conclusion

This paper presented an uncertainty analysis for a vision framework using the primitives proposed by [8], and the scene description in terms of inter-primitives relation discussed in [1]. In a first part we discussed how image and calibration uncertainty propagates during the reconstruction process. This result, although classic in nature (e.g., [2]), allowed us to formalise the peculiarities of the uncertainty space that stems from our use of local line descriptors (mainly its strong dependence on 2D orientation). This analytic

formulation of uncertainty propagation was demonstrated to be accurate by Monte Carlo simulations.

The second and most important part of this paper considers inter-primitives geometric relations, focusing on the cases of distance and coplanarity. In [1] it was discussed that such relations form a good base for the interpreting visual information, and confronting it with available prior geometric information about the scene. The 3D-primitives’ uncertainties computed in the first part were used to design alternative formulations of those relations that take parameters’ uncertainty into account. The new formulations were shown to detect geometric relations in a more robust fashion than the naive Euclidian ones, and across wide ranges of depth.

In this paper we derived an analytic formulation for uncertainty propagation during the stereo-reconstruction of local line primitives. Although the derivation presented here is specific to the representation proposed in [8], it can be easily adapted to other line-based features. The advantage of an explicit analytic formulation of the uncertainty is, it allows us to accurately model the whole complexity of the uncertainty space. Estimating such a high dimensional space by Monte Carlo simulation would be impractical.

In a second part we used these uncertainties to define relations between reconstructed 3D-primitives. The relations are a way to apply prior geometrical knowledge about the scene onto the reconstructed 3D representation. Such relations need to allow for a certain imprecision in the 3D-primitives, imprecision that is itself a function of the parameters thereof. Therefore, in order to define inter-primitives relations in a robust manner, we need to take into account the uncertainty defined in the first part of this paper. We proposed covariance-adapted criteria for distance and coplanarity, and we have shown that these criteria were more robust to differences in depth than naive Euclidian definitions.

## References

- [1] E. Baseski, N. Pugeault, S. Kalkan, D. Kraft, F. Wörgötter, and N. Krüger. A scene represen-

- tation based on multi-modal 2d and 3d features. In *ICCV '07 Workshop on 3D Representation for Recognition (3dRR-07)*, 2007. 1, 8
- [2] J. C. Clarke. Modelling uncertainty: A primer. Technical report, Department of Engineering Science, Oxford University, 1998. 2, 3, 4, 8, 9, 10
- [3] G. Csürka, C. Zeller, Z. Zhang, and O. Faugeras. Characterizing the Uncertainty of the Fundamental Matrix. *Computer Vision and Image Understanding*, 68(1):18–36, 1997. 4
- [4] H. F. Durrant-Whyte. Uncertain Geometry in Robotics. *IEEE Journal of Robotics and Automation*, 4(1):23–31, 1988. 2, 3
- [5] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993. 2, 3, 4, 10
- [6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 3, 4
- [7] S. Heuel and W. Förstner. Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. In *CVPR '01*. IEEE, 2001. 3, 4
- [8] N. Krüger, N. Pugeault, and F. Wörgötter. Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information (also available as technical report (2007-4) of the robotics group maersk institute, university of southern denmark). submitted. 2, 3, 4, 8
- [9] R. Mandelbaum, G. Kamberova, and M. Mintz. Stereo depth estimation: a confidence interval approach, 1998. 3
- [10] E. Maxwell. *General homogeneous coordinates in space of three dimensions*. Cambridge University Press, 1951. 10
- [11] J. J. Rodríguez and J. K. Aggarwal. Quantization error in stereo imaging. In *Proceedings of the CVPR*, 1988. 3
- [12] F. Shevlin. Analysis of orientation problems using Plücker lines. *International Conference on Pattern Recognition, Brisbane*, 1:65–689, 1998. 10
- [13] A. Verri and V. Torre. Absolute depth estimate in stereopsis. *Journal of Optical Society of America*, 3:297–299, 1986. 2
- [14] L. B. Wolff. Accurate measurements of orientation from stereo using line correspondence. 1989. 3, 4

## A Error propagation

**Image point back-projection covariance** The intersection of the optical ray back-projected from an image point with the plane at infinity is computed as follows:

$$\mathbf{D} = \mathbf{P}^{-1}\mathbf{m} = f_D(u, v, 1, p_{11}, \dots, p_{33}) \quad (10)$$

and therefore, according to Eq. 3 we have

$$\mathbf{\Lambda}_D = \nabla f_D \cdot \begin{pmatrix} \mathbf{\Lambda}_m & \\ & \mathbf{\Lambda}_{\mathbf{P}^{-1}} \end{pmatrix} \cdot \nabla f_D^\top \quad (11)$$

where  $\nabla f_D$  stands for the Jacobian of a  $\mathbb{R}^{12}$  to  $\mathbb{R}^3$  mapping (therefore a  $3 \times 12$  matrix), and is defined as follows:

$$\nabla f_D = \begin{pmatrix} P_{*1}^{-1} & P_{*2}^{-1} & 0_3^\top & \begin{pmatrix} \mathbf{m}^\top & & \\ & \mathbf{m}^\top & \\ & & \mathbf{m}^\top \end{pmatrix} \end{pmatrix} \quad (12)$$

where  $\mathbf{P}_{*i}^{-1}$  refers to the  $i^{th}$  column vector of the matrix  $\mathbf{P}^{-1}$ .

The vector  $\mathbf{D}$  is then normalised as in appendix A.

**Covariance of the cross product** The cross product of two vector propagates uncertainties according to the following formula (see [2]):

$$\mathbf{\Lambda}_{C \times D} = \nabla(C \times D) \cdot \begin{pmatrix} \mathbf{\Lambda}_C & \\ & \mathbf{\Lambda}_D \end{pmatrix} \cdot \nabla(C \times D)^\top \quad (13)$$

where the Jacobian of the cross product is defined as follows

$$\nabla(C \times D) = \begin{pmatrix} -[D]_\times & [C]_\times \end{pmatrix} \quad (14)$$

In this equation,

$$[\mathbf{A}]_\times = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \quad (15)$$



correspond to the matrix formulation of the cross product with the vector  $\mathbf{a} = (a_1, a_2, a_3)^\top$ , such that  $[\mathbf{A}]_\times \cdot (x_1, x_2, x_3)^\top = \mathbf{a} \times (x_1, x_2, x_3)^\top$ .

**Vector normalisation covariance** Any vector  $\mathbf{v}$  with a covariance matrix  $\mathbf{\Lambda}_\mathbf{v}$  is collinear with a normalised vector  $\bar{\mathbf{v}}$  with a covariance matrix:

$$\mathbf{\Lambda}_{\bar{\mathbf{v}}} = \nabla_{\text{norm}} * \mathbf{\Lambda}_\mathbf{v} * \nabla_{\text{norm}}^\top \quad (16)$$

where  $\nabla_{\text{norm}}$  is the Jacobian of the normalisation operation, defined as follows:

$$\nabla_{\text{norm}} = \frac{1}{\|\mathbf{v}\|^3} \cdot \begin{pmatrix} v_2^2 + v_3^2 & -v_1 v_2 & -v_1 v_3 \\ -v_1 v_2 & v_1^2 + v_3^2 & -v_2 v_3 \\ -v_1 v_3 & -v_2 v_3 & v_1^2 + v_2^2 \end{pmatrix} \quad (17)$$

**Conversion from homogeneous to Euclidian coordinates** The conversion from homogeneous coordinates to Euclidian is obtained by the following formula:

$$g : \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \rightarrow \begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \\ \frac{z}{w} \end{pmatrix} \quad (18)$$

According to [2], the covariance of the resulting dimension 3 vector is given by the following formula:

$$\mathbf{\Lambda}_\mathbf{M} = \nabla g \cdot \mathbf{\Lambda}_{\tilde{\mathbf{M}}} \cdot \nabla g^\top \quad (19)$$

where  $\tilde{\mathbf{M}}$  is the homogeneous coordinate vector,  $\mathbf{M}$  is the 3D Euclidian coordinate vector, and  $\nabla g$  is the Jacobian of the conversion (see [2]):

$$\nabla g = \frac{1}{w^2} \begin{pmatrix} w & 0 & 0 & -x \\ 0 & w & 0 & -y \\ 0 & 0 & w & -z \end{pmatrix} \quad (20)$$

## B Uncertainty propagation during reconstruction

**Uncertainty of the back-projected ray** Consider a camera with an optical centre  $\mathbf{C}$  and a projection matrix  $\tilde{\mathbf{P}}$  (the tilde denotes an homogeneous

formulation). The optical centre of a camera is given by its projection matrix (see, e.g., [5]), and therefore its covariance is also null:  $\mathbf{\Lambda}_\mathbf{C} = \mathbf{0}_{3 \times 3}$ . Then, all 3D points that could possibly be projected by this camera on an image point  $\mathbf{m}$  lie on the line in space defined by the Plücker coordinates  $\mathcal{L} = (\boldsymbol{\nu}, \boldsymbol{\mu})^\top$ . In the Plücker representation of lines in space,  $\boldsymbol{\nu}$  is the line's orientation vector, and  $\boldsymbol{\mu}$  is called its *moment*, and satisfies  $\mathbf{m} \cdot \boldsymbol{\nu} = \boldsymbol{\mu}$ ,  $\forall \mathbf{m} \in \mathcal{L}$ . See [12] for a discussion of Plücker's formulation advantages. From, e.g., [10], we have

$$\mathcal{L} = (\mathbf{D}, \mathbf{C} \times \mathbf{D}) \quad (21)$$

where  $\mathbf{D}$  is the back-projection of the point  $\mathbf{m}$  on the plane at infinity — see appendix A. According to Eq. (2), this line's covariance is approximated by the following  $6 \times 6$  block diagonal matrix:

$$\mathbf{\Lambda}_\mathcal{L} = \begin{pmatrix} \mathbf{\Lambda}_\nu = \mathbf{\Lambda}_\mathbf{D} & \\ & \mathbf{\Lambda}_\mu = \mathbf{\Lambda}_{\mathbf{C} \times \mathbf{D}} \end{pmatrix} \quad (22)$$

where  $\mathbf{\Lambda}_\mathbf{C}$  and  $\mathbf{\Lambda}_\mathbf{D}$  are  $3 \times 3$  covariance matrices.

**Uncertainty of the back-projected plane** Considering the same camera set-up, all 3D-points that project onto a line containing the point  $\mathbf{m}$  and with an orientation  $\mathbf{t}$ , lie on a 3D plane  $\mathcal{P} = (\boldsymbol{\eta}^\top, -h)$ , where  $\boldsymbol{\eta}$  is the normal to the plane  $\mathcal{P}$ , and  $h$  is the Hesse distance (the smallest distance from the origin to a point on the plane). From classical geometry, we know that a plane in space can be computed from 3 non-collinear points, or alternatively from one point and two non-collinear vectors. In this case, we use the optical centre  $\mathbf{C}$ , the 2D-primitive's position  $\mathbf{m}$  back-projection at infinity  $\mathbf{D}$  (see appendix A), and the 2D-primitive's orientation vector  $\mathbf{t}$  to compute the plane's coordinates:

$$\mathcal{P} = \begin{pmatrix} \mathbf{T} \times \mathbf{D} \\ \mathbf{C}^\top \cdot (\mathbf{T} \times \mathbf{D}) \end{pmatrix} = f_\mathcal{P}(\mathbf{C}, \mathbf{D}, \mathbf{T}) \quad (23)$$

where  $\mathbf{T} = \mathbf{P}^{-1} \cdot \mathbf{t}$  is a vector lying on the plane  $\mathcal{P}$ .

Therefore the plane's covariance is the  $4 \times 4$  matrix:

$$\mathbf{\Lambda}_\mathcal{P} = \nabla f_\mathcal{P} \cdot \begin{pmatrix} \mathbf{\Lambda}_\mathbf{C} & & \\ & \mathbf{\Lambda}_\mathbf{D} & \\ & & \mathbf{\Lambda}_\mathbf{T} \end{pmatrix} \cdot \nabla f_\mathcal{P}^\top \quad (24)$$

where  $\nabla f_{\mathcal{P}}$  is the  $4 \times 9$  Jacobian

$$\nabla f_{\mathcal{P}} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & -[\mathbf{T}]_{\times} & [\mathbf{M}]_{\times} \\ \mathbf{M}_{\infty} \times \mathbf{T} & \mathbf{T} \times \mathbf{C} & \mathbf{D} \times \mathbf{C} \end{pmatrix} \quad (25)$$

**Intersection between a line and a plane** The line  $\mathcal{L} = (\boldsymbol{\nu}^{\top}, \boldsymbol{\mu}^{\top})$  intersect with the plane  $\mathcal{P} = (\boldsymbol{\eta}^{\top}, -h)$  at the homogeneous point:

$$\tilde{\mathbf{M}} = \begin{pmatrix} \boldsymbol{\eta} \times \boldsymbol{\mu} + h\boldsymbol{\nu} \\ \boldsymbol{\eta} \cdot \boldsymbol{\nu} \end{pmatrix} = f_{\tilde{\mathbf{M}}}(\boldsymbol{\nu}^{\top}, \boldsymbol{\mu}^{\top}, \boldsymbol{\eta}^{\top}, h) \quad (26)$$

The uncertainty of this computation is computed by applying Eq.(2) to Eq.(26):

$$\boldsymbol{\Lambda}_{\tilde{\mathbf{M}}} = \nabla f_{\tilde{\mathbf{M}}} \cdot \begin{pmatrix} \boldsymbol{\Lambda}_{\mathcal{L}} & \\ & \boldsymbol{\Lambda}_{\mathcal{P}} \end{pmatrix} \cdot \nabla f_{\tilde{\mathbf{M}}}^{\top} \quad (27)$$

There,  $f_{\tilde{\mathbf{M}}}$  is the intersection function, defined as follows:

$$f_{\tilde{\mathbf{M}}}(\boldsymbol{\nu}^{\top}, \boldsymbol{\mu}^{\top}, \boldsymbol{\eta}^{\top}, h) = \begin{pmatrix} \boldsymbol{\eta} \times \boldsymbol{\mu} + h\boldsymbol{\nu} \\ \boldsymbol{\eta} \cdot \boldsymbol{\nu} \end{pmatrix} \quad (28)$$

and its Jacobian  $\nabla f_{\tilde{\mathbf{M}}}$  is the  $4 \times 10$  matrix:

$$\nabla f_{\tilde{\mathbf{M}}} = \begin{pmatrix} h \cdot \mathbf{I}_{3 \times 3} & \nabla(\boldsymbol{\eta} \times \boldsymbol{\mu}) & \boldsymbol{\nu} \\ \boldsymbol{\eta}^{\top} & \mathbf{0}_3^{\top} & \boldsymbol{\nu}^{\top} & 0 \end{pmatrix} \quad (29)$$

This yields 3D homogeneous points, that can be transformed to 3D Euclidian coordinates — see appendix A.

**Intersection of two planes** The 3D-primitive's orientation is reconstructed by intersecting the two back-projected planes  $\mathcal{P}^l = (\boldsymbol{\eta}^{\top}, -h)$  (back-projected from the left image) and  $\mathcal{P}^r = (\boldsymbol{\eta}'^{\top}, -h')$  (back-projected from the right image):

$$\mathcal{L} = l(\mathcal{P}^l, \mathcal{P}^r) = \begin{pmatrix} \boldsymbol{\eta} \times \boldsymbol{\eta}' \\ h'\boldsymbol{\eta} - h\boldsymbol{\eta}' \end{pmatrix} \quad (30)$$

because we are only interested in the line's orientation, we can disregard its moment (three last columns and rows of the block matrix). Therefore, the orientation vector's covariance is given by the cross product of the two planes' normals:

$$\boldsymbol{\Lambda}_{\mathcal{L}} = \nabla(\boldsymbol{\eta} \times \boldsymbol{\eta}') \cdot \begin{pmatrix} \boldsymbol{\Lambda}_{\boldsymbol{\eta}} & \\ & \boldsymbol{\Lambda}_{\boldsymbol{\eta}'} \end{pmatrix} \cdot \nabla(\boldsymbol{\eta} \times \boldsymbol{\eta}')^{\top} \quad (31)$$

We obtain  $\boldsymbol{\Lambda}_{\boldsymbol{\eta}}$  from the plane's covariance matrix  $\boldsymbol{\Lambda}_{\mathcal{P}}$  three upper rows, as computed in Eq.(24). The cross product error propagation is given in appendix A.

Therefore Eq. (28) and (31) predict a reconstructed 3D-primitive's localisation and orientation uncertainty as function of several parameters: 2D-primitive extraction, stereo-match and projection matrices uncertainties on the one hand, and 2D-primitives' position and orientation on the other.