Scene analysis of unstable video flows – using multiple retina channels and attentional methods

Anna Lázár[1], Karl Pauwels[2], Marc Van Hulle[2] and Tamás Roska[1,3]

[1]Peter Pazmany Catholic University, Faculty of Information Technology, Prater utca 50/a,
1083 Budapest, Hungary*
[2]Laboratorium voor Neuro- en Psychofysiologie, K.U.Leuven, Belgium
[3]MTA SZTAKI, Computer and Automation Research Institute, Hungarian Academy of Sciences,
Kende u. 13-17, 1111 Budapest, Hungary

Abstract: The present work describes a bio-inspired approach designed to solve some tasks raised in the 'Bionic Eyeglass Project' [1], which aims to help the everyday life of blind or visually impaired people. The purpose of this approach is to provide a specific kind of information or to determine the region of interest ("ROI") in a low-resolution and unstable video flow recorded with a mobile phone by the visually impaired person. In the present paper we introduce a new stabilization method and three different tasks are resolved: firstly, locating LED indicators, secondly, locating traffic signs, and thirdly, deciding whether there is any switched-on lamp in a room. The test database has been made out of real-life scenes. We provide detailed evaluation results referring to the execution of those tasks.

Our descriptive context refers to the recently modeled mammalian retina channel decomposition [2]. Using it we can avoid - at least partially - the classical difficulty that image processing algorithms nowadays face, namely that the intensity or color values of the same object largely depend on the actual lighting conditions. A further difficulty referring to the lamp-detection is that the solution has to be completely independent of the input's actual brightness. The method we introduce relies only on a single retina channel and achieves a very high accuracy: the ratio of the correct answers is around 99%. The other two tasks are to carry out an approximately real-time ROI-detection algorithm based solely on image information from an unstable low-resolution video-flow containing complex real-life scenes with unconstrained lighting conditions. The accuracy of the introduced methods is around 80%.

We also make use of a stabilization algorithm designed especially for this project. In order to yield the desired information, we process channel-data as well as saliency maps. The presented method can be useful in a variety of other application areas.

Keywords: Video flow stabilization, retina channel, CNN, Cellular Wave Computing, region of interest, saliency map

1) Introduction:

Although an encouraging progress has been already achieved concerning retinal prostheses, the everyday usage of the related techniques still seems to be remote. Until then, and in many other situations, different methods should help the everyday life of the blind or visually impaired people. This paper describes a bio-inspired method aiming to locate those

* correspondence e-mail address: lazar@digitus.itk.ppke.hu

regions in the visual scene that, with a high probability, contain important information for visually impaired people – that is: to define the Region of Interest (ROI) in an unstable, low resolution video input. For this purpose, we also use the mammalian retina channel decomposition. After locating the regions that include the required information, according to the actual task, different pattern-detection or object recognition algorithms can be used. Similar situations arise at some other blind navigation tasks, in robotics, and other applications.

In this stage of the project, the input comes from a mobile phone's video camera in 176 x 144 pixel resolution, but the phone is now being extended by a Cellular Visual Microprocessor[*]. The diversity of interesting tasks as well as the construction of the required database has been compiled with the help of members of the 'Hungarian National Association of Blind and Visually Impaired People' [1]. In this paper we present efficient new algorithms for:

- Finding light sources (lamps) – this task (although it seems to be a trivial 'problem' for a person with normal vision), could prevent annoyance for visually impaired people, for example, by preventing the lamps to remain switched-on for weeks after a guest.

  Here, the most important criterion is that the solution has to be independent from the input's actual brightness, that is, the accuracy should be the same in the case of a sun-drenched and a dark room.
- Locating LED indicators (in real-life indoor and outdoor scenes)
- Finding traffic signs in real-life street scenes.

  The main purpose of these two latter tasks are to realize a fast method that locates the areas which contain the traffic signs / LED indicators with high probability, on complex real-life outdoor scenes. Subsequently, a classifier algorithm has to analyze only the located ROIs instead of the whole input, which can fasten up the whole process significantly. The main difficulties derive from the instability of the by-default bad-resolution input, the unconstrained lighting conditions, and from the *variety* of the possible inputs.

The algorithms' main functional components are: video stabilization, retina channel decomposition, (or "low-level feature extraction"), and saliency map generation.

A summarizing flow chart can be seen on figure 1.

[*] This visual microprocessor is the Q-Eye in the Eye-RIS system: www.anafocus.com

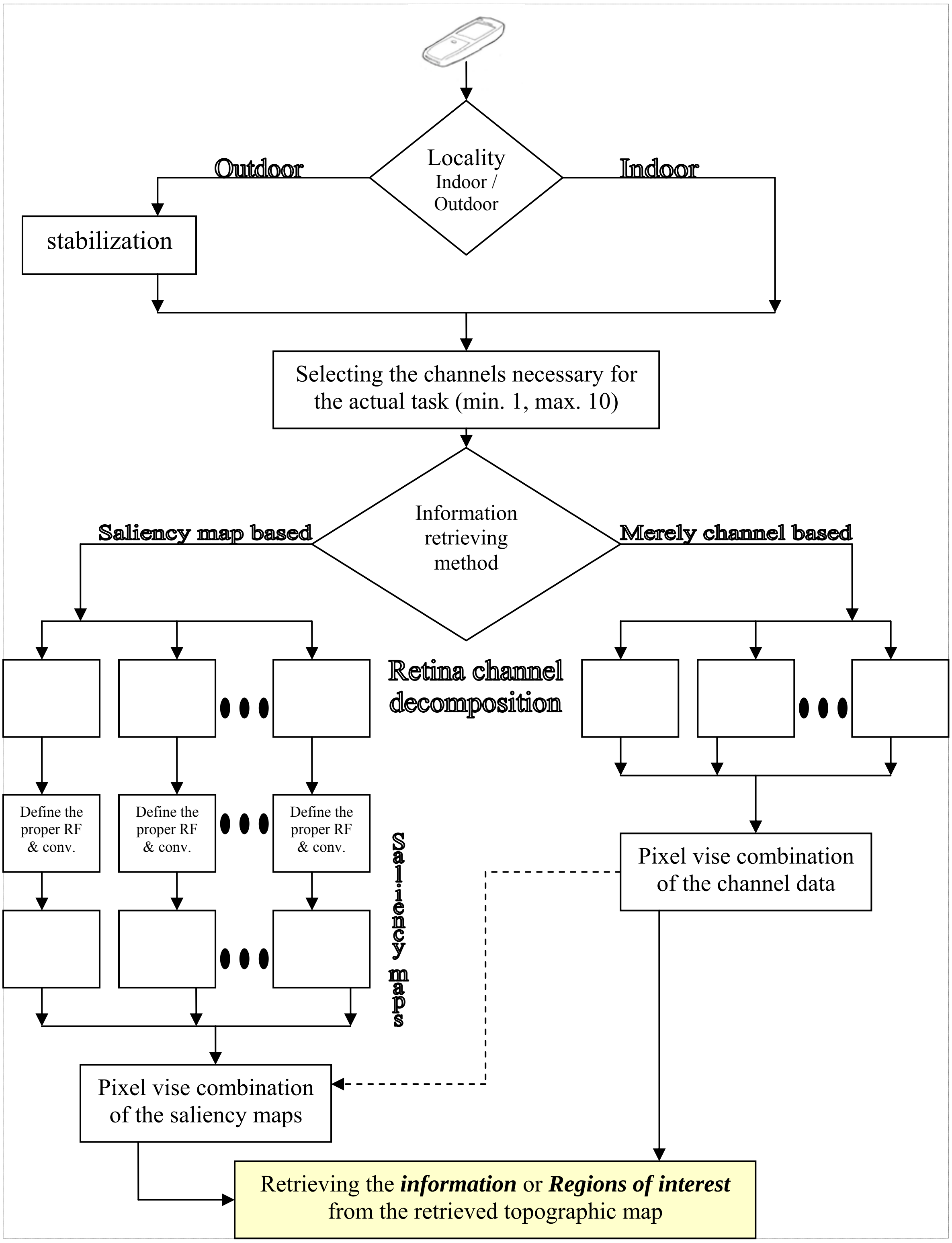
Locality
Indoor /
Outdoor
Outdoor
Indoor
stabilization
Selecting the channels necessary for the actual task (min. 1, max. 10)
Information retrieving method
Saliency map based
Merely channel based
Retina channel decomposition
Define the proper RF & conv.
Define the proper RF & conv.
Define the proper RF & conv.
Saliency maps
Pixel vise combination of the channel data
Pixel vise combination of the saliency maps
Retrieving the information or Regions of interest from the retrieved topographic map

Figure 1. The flow chart of the proposed method. The input is a strongly unstable, low resolution video flow coming from a mobile phone's camera held by a visually impaired person. The output can be

- *Regions of interest* (e.g. locations of LED indicators, traffic signs), or
- *Specific information* (e.g. is there any switched-on lamp).

The dashed line shows an optional information combination step (raised in the task of locating traffic signs, where retina channel data and saliency map data had been combined, see section 2.3.1)

2) The proposed method:

Figure 1 summarizes the algorithm. The input of the *whole* process is an image flow taken by a mobile phone extended with a Cellular Visual Microprocessor, and the output consists of audio information for the person using the equipment. The present paper does not deal with the methodology of transformation of the demanded information into audio format, but with the problem of *locating* the demanded information within a video flow.
These steps are described in detail in the following sections.

2.1) Stabilizing the input frame

Image flows provided by a camera held by a blind walking person are usually extremely noisy and unstable, often accompanied by fast, unexpected camera motions. The recording equipment (camera) can be

- rotated
- shifted in the vertical and horizontal direction, and
- transported in the direction of motion.

Additionally, often the picture's main objects shift significantly from one frame to another, e.g. during turning around.

The goal of the image stabilization step is to keep the steady objects (e.g. buildings) in the same pixel positions, while the moving objects (for example the pedestrians) can change position.

It is useful to define the transformation-parameters between *adjacent* frames, instead of estimating the difference between the reference frame and the actual frame. In this manner, it is possible to trace bigger deformations throughout longer frame-series. Then, the calculated transformation 'inherits' from frame to frame, as follows:

If the actual reference frame is the i^{th} one, then the $\boldsymbol{P}_{i+k,i}$ vector contains the transformation parameters between the actual frame $i+k$ and the reference frame i. In the next step, the vector $\mathbf{P}_{i+k+1,\, i+k}$ is calculated, which contains the transformation values between the actual adjacent frames: $i+k+1$ and $i+k$. Then $\boldsymbol{P}_{i+k+1,\, i} = \boldsymbol{P}_{i+k,\, i} \oplus \boldsymbol{P}_{i+k+1,\, i+k}$ will be updated, and will comprise of the differences accumulated throughout the $k+1$ frames that have been captured since the last reference-frame updating.

Figure 2 depicts the flow chart of the stabilization. (See also [3, 4]) The key element in it, is how the transformation parameters are defined (we highlighted this step with a bit darker shade on the diagram).

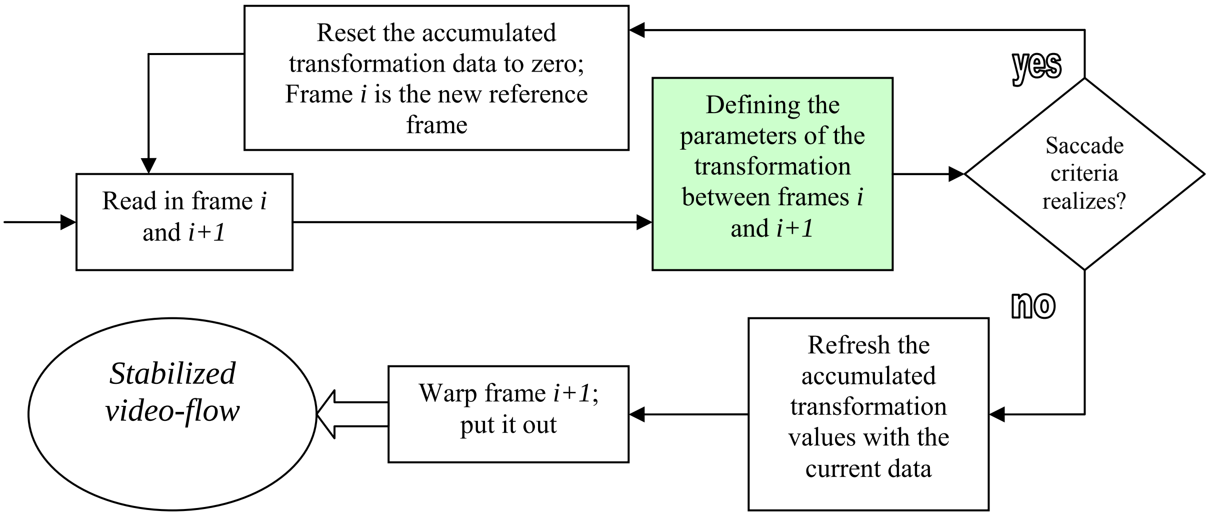
Reset the accumulated transformation data to zero; Frame *i* is the new reference frame
Read in frame *i* and *i+1*
Defining the parameters of the transformation between frames *i* and *i+1*
Saccade criteria realizes?
yes
no
Refresh the accumulated transformation values with the current data
Warp frame *i+1*; put it out
Stabilized video-flow

Figure 2: The flow chart diagram of the stabilization. The input (left hand side, top of the picture) is an unstable video-flow coming from a mobile phone's camera. The output of this algorithm is the stabilized video flow (left hand side, bottom of the picture; Details in section 2.2). The goal is to keep the steady objects (e.g. buildings) in the same pixel positions, while the moving objects (e.g. pedestrians) can change position.

We have estimated the transformation parameters between frames *i* and *i+1* as follows:

Let *f* denote the *image intensity function*, which is the intensity value at *x-y* coordinate position at time *t*. [5] (*f* = *f(x, y, t)*) Thus:

$$-\frac{\partial f}{\partial t}=\frac{\partial f}{\partial x}u+\frac{\partial f}{\partial y}v \qquad (1)$$

where $\partial f/\partial x$ and $\partial f/\partial y$ are the spatial gradients in *x* and *y* directions, and $\partial f/\partial t$ is the time gradient. Since these quantities are measurable, using equation (1), *u* and *v* can be defined, which are the velocities in *x* and *y* directions, respectively. (The unit can be pixel/frame.) Since the frames are not only translated, *u* and *v* differ for every single pixel position. To estimate these, we have used an affine transformation model, which can handle translation, scaling, rotation and shear: [3]

$\begin{bmatrix} \mathrm{u} \\ \mathrm{v} \end{bmatrix} = \begin{bmatrix} \mathrm{a}_0 \\ \mathrm{b}_0 \end{bmatrix} + \begin{bmatrix} \mathrm{a}_1 & \mathrm{a}_2 \\ \mathrm{b}_1 & \mathrm{b}_2 \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix}$, where $\begin{bmatrix} a_0 \\ b_0 \end{bmatrix}$ defines the translation in *x* and *y* directions, and $\begin{bmatrix} \mathrm{a}_1 & \mathrm{a}_2 \\ \mathrm{b}_1 & \mathrm{b}_2 \end{bmatrix}$ describes the scaling, rotation and shear.

Finally, for every frame, the following matrix-equation has to be solved, where the vector ***P*** contains the six parameters: a_0, a_1, a_2, b_0, b_1, b_2, which describes the transformation.

$$P = A^{+} \cdot (-I_t)$$

A and $\boldsymbol{I_t}$ contain the intensity gradients and the general coordinates, the sign '+' denotes pseudo inverse. For the detailed definitions and derivations, see the Appendix.

2.2) Retinal output channels

This section describes the background and the usage of the retinal output channels. We use:

- Seven "spatio-temporal" channels (which are the 'Transient', Local Edge Detector, 'Bistratified', 'Alpha', 'Beta', 'Delta' and the 'Polar' channels) [6,2].
- Three color channels (which are the Intensity and the two color opposition channels): ("R" is the red value in the RGB system, "G" is the green, and "B" is the blue) [7]:
    - Red-Green opponent = R-G
    - Blue-Yellow opponent = B – (R + G) / 2
    - Intensity = 0.18*R + 0.81*G + 0.01*B

Systems designed to retrieve information of visual input often use some kind of "low level visual feature-extraction" before further processing. The main reason for this is that, otherwise, the appearance of the different objects bewilderingly depends on the illumination, perspective, scaling and other actual circumstances. That is, the same object can have completely different intensity- or color values according to some accidental conditions.

The mammalian nervous system also applies this 'trick': it dissolves the visual information simultaneously in approximately a dozen different specialized channels, each coding different low level features [8]. Their exact function is defined by certain ganglion-cell types [6]. These spatio-temporal channels arise in the retina and persist up to the high brain areas, while performing several processing steps.

Our model includes all the ten, biologically measured [6] and artificially modeled [2] channels. This means that we take into account three time-independent (one intensity and two color opposition channels) as well as seven "spatio-temporal" channels. Although some basics are known, the method specifying how the colors are being processed is mainly undiscovered. From the viewpoint of the realization, we have applied what is known by the present: the Intensity channel is calculated as: 0.18*R + 0.81*G + 0.01*B [7], the red-green opponent channel is the difference between the red-value and the green-value, while the blue-yellow is calculated like "blue – (red + green) / 2", as described above.

The seven parallel pathways have the same design; they only differ in the parameters that determine their distinct spatio-temporal characteristics [6, 2, 11]. For these channels, a neuromorphic model has been created, based on Cellular Neural/Nonlinear Network (CNN) architecture [9, 10]. In this, the basic processing principles of the retina have been kept, but in a simplified form. We mention that choosing a CNN simulator has been plausible because of the striking similarity between the CNN structure and the living retina layers.
The exact retina channel model we have taken as a basis is detailed in [2].

Figure 3 shows a snapshot of the ten retina channels for a natural scene. Interesting to notice, that only half of the channels' function is known, in the sense that the aim of the process of the remaining five channels could not be formulated explicitly, at least up to present. Thus: the Transient channel filters out everything that is in motion and eliminates all the steady parts [11], the Intensity channel codes the intensity, as its' name presumes, the Blue-Yellow and the Red-Green are color opposition channels, and finally the 'LED' is for 'Local Edge Detector', whose role is to emphasize the edges. The function of the Bistratified, Polar, Alpha, Beta and Delta are unknown, they are primary named after the ganglion cells that code them.

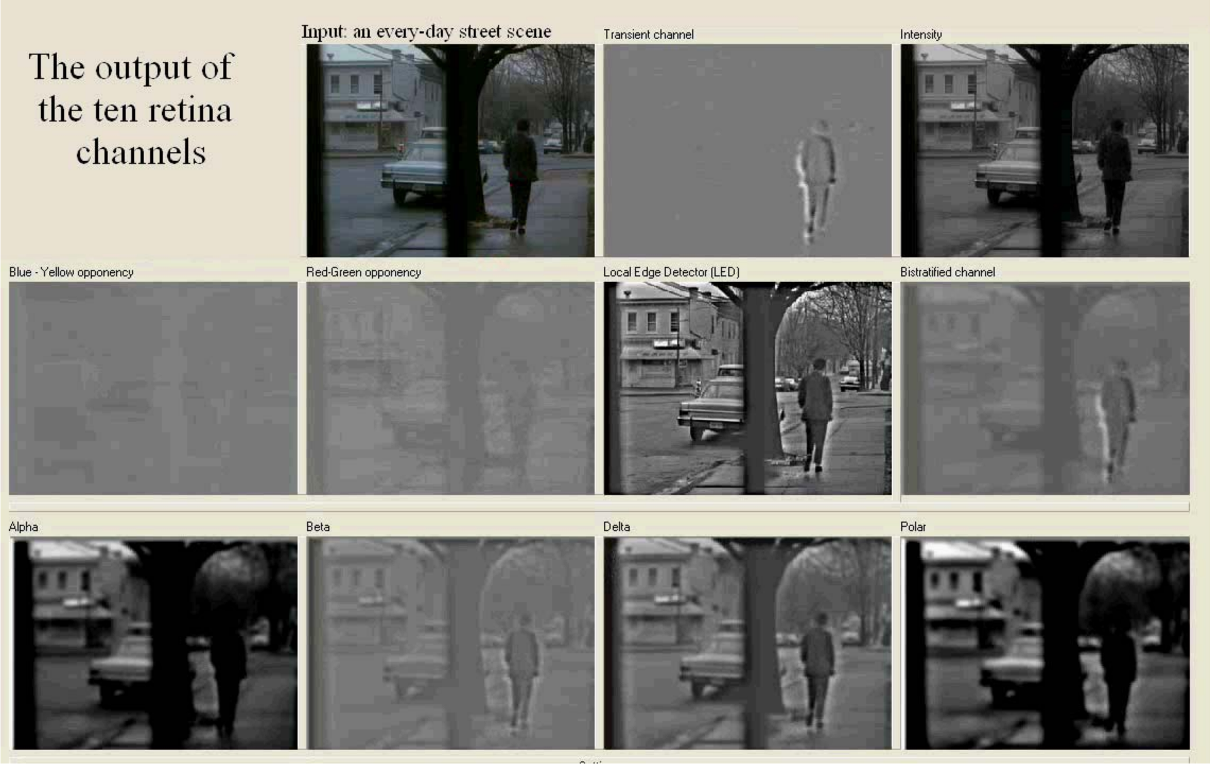

Figure. 3. An example for the ten retina channels. The input image (first picture) is processed by ten different pathways resulting in ten ganglion-cell types which form the ten retina channels [6]. The second picture in the first row (next to the input image) is the output of the "transient" channel which filters out the mobile parts of the visual scene and removes all the steady sections: at this moment the walking girl triggers the only response. Normally this is one of the 'strongest' channels. The last image in the first row depicts the output of the "intensity" channel. In the second row we can see the blue-yellow- and the red-green contrast channels (these are the color opposition channels), the LED (local edge detector) and the "bistratified" channels. The functions of the channels depicted in the third row (alpha-, beta-, delta- and polar) are unknown at present (in the sense that we can not 'phrase' their function), as well as the bistratified channel's task. Note that many of these channels can not be represented by still images or still inputs.

2.3) Receptive fields and saliency maps for the depiction of Regions of Interest

Saliency maps are two-dimensional, scalar maps of the physical world, whose activity topographically represent visual conspicuity. In most attentional models, every channel creates its' own saliency map, which is feature-dependent (the feature refers to what the given channel codes). Those saliency maps are afterwards usually unified into a final "master" map, which is thus feature-independent [12]. This process can be used effectively in different real applications as well.

In nature, saliency is "calculated" via receptive fields (RF), in which neurons are organized into concentric circles: a central and a peripheral part, which respond antagonistically. If the central part of an ON-center - OFF surrounding RF is stimulated with light, the RF's response will increase, while if the light falls onto the surrounding part, then the response will decrease. If both parts are exposed to light, then there will be no change in the ganglion cells' response [7].

Practically, from an engineering viewpoint, a saliency map is a retina channel output (or the result of the 'low level visual feature extraction') convolved with a receptive field. RFs can be represented in matrix form.

As an example of the used receptive fields, we show how we have determined the optimal RF in the task of finding traffic signs. The outcome is depicted on figure 4.

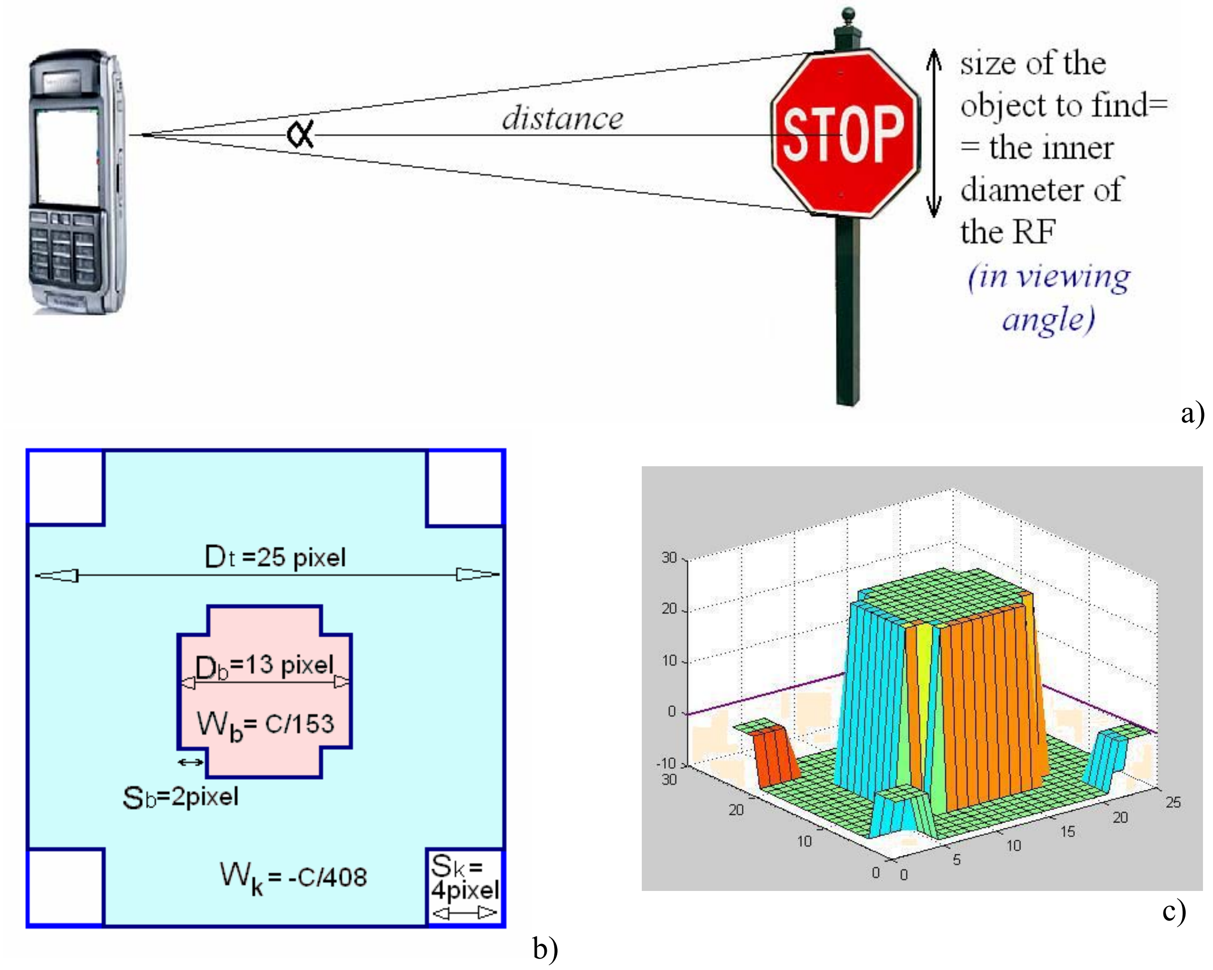

Figure 4. Receptive field adjusted for the task of finding traffic signs. Figure ***a)***: the inner diameter of the receptive field and the size of the searched object should be the same in viewing angle. This criterion helps to determine the size of the RF. ***b)*** and ***c)***: the resultant; On ***c)*** the height and the depth are proportional to the weights, which have – due to the antagonistical behaviour of the RF's inner and outer part – opposite sign. The zero level is emphasized with purple line.

- The *size* of the receptive field is determined as follows:
  - from one hand, the viewing-angle of the mobile phone is ~45°, which occupies 176 pixels. This means, that roughly 3.9 pixels cover 1°.
  - from the other hand, on the video flow, the size of an object depends on its distance. Namely, its' size in viewing angle is $\mathrm{tg}\,{}^{\alpha}\!/_{2} = \frac{\text{radius\_of\_the\_object}}{\text{dis}\tan\text{ce}}$ (figure 4 a). So, if we want to set the sensing-distance of a 45 cm diameter (0.225 m radius) traffic sign for ~7-8 m, then it covers a bit more than 3.3°. Thus, the optimal inner diameter of the receptive field is 13 pixels. (figure ***4 (a)*** and (***b)***)
  - The outer size of the RF has been adjusted according to 'real' RFs, in which the inner part covers around the half of the whole RF in viewing angle.
- The *values* of the matrix have to satisfy the criteria of
  - giving maximal response if antagonistic stimuli hit the RF's inner and outer area
  - giving no answer if the input image-region contains equal values – that is, in case of uniform lighting.

  The maximal value is arbitrary, since it is only a constant multiplier ("C" on figure ***4 (b)***).

2.3.1) Locating traffic signs

The main purpose of the present algorithm is to realize a fast method locating the areas which contain traffic signs with high probability, on complex real-life outdoor scenes. The main difficulties derive from the instability of the input – which has by default bad resolution – and from the fact that the lighting circumstances can vary on a wide range.

Traffic signs – due to their color and shape design - can effectively be detected by circle-shaped receptive fields on color opposition channels. For solving this task, we have used the RF determined in the previous section (figure 4 b) on the Blue-Yellow color opposition channel. According to the experiments, rooftops and building walls often effectuate salient areas with the blue sky, something that can lead to false results. In order to avoid these errors, we have applied the Delta channel's data as well: since it gives a vivid response on light sources, only those regions have been taken into account, where this channel has given a smaller response than a given threshold.

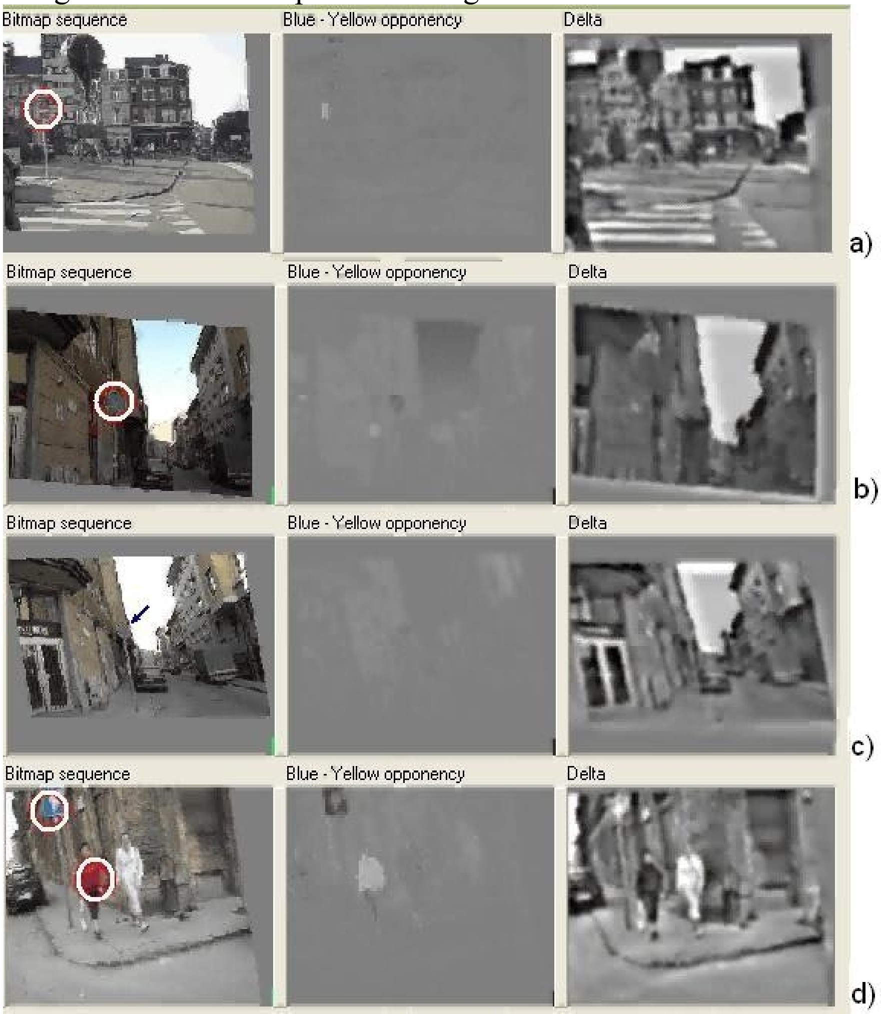

Figure 5: Some typical frames from the test database we have used to evaluate the task aiming to locate traffic signs. In all the four rows, the left-most image is a frame from the input video flow with the areas identified as traffic signs (white circles). The other two images in each row are the corresponding outputs of the used channels: the middle ones are the Blue-Yellow color opposition channels' output and the right ones are the response of the Delta channel. **a)** and **b)** are examples for correct results. **c)** The prime cause of the false negative answers (when the sign is not located) was due to the loss of the color information, which happened when the sign was in shadow. On figure **c)** the blue arrow points to a traffic sign being in shadow. (These signs are difficult to see even with "pure eyes") From closer they can be identified: **b)** is the same as **c)** from a few meters nearer). **d)** depicts the typical reason for false positive results: vivid colors with the 'appropriate size'.

The input frames are distorted because of the stabilization. Table I and II indicate the test results.

Figure 5 shows some typical frames from the test database. The distortions of the input frames are due to the stabilization method. Important to note, that this method does not exploit any additional information or knowledge (for example, that traffic signs are primarily expected in a given height), thus with the guidance of the equipment the results can be further improved. According to the test results, the main error sources have been: from one hand, *shadow*, which leads to false negative results because of the loss of color information (figure 5 c), and from other hand, objects with 'appropriate' size and vivid colors, which lead to false positive results (figure 5 d). Important to note, that – because of the lack of a commonly accepted test database for these problems - the evaluated information significantly depends on the test database. Tables I and II show our test results.

(frame percentage)	*Correct answer*	*False answer*
There *is* traffic sign on the input video frame (total 502 frames)	**73.7%** (370 frames out of 502 frames)	**26.3%** (132 frames out of 502 frames)
There is *no* traffic sign on the input video frame (total 414 frames)	**95.4%** (395 frames out of 414 frames)	**4.6%** (19 frames out of 414 frames)
Total (916 frames)	**83.5%** (765 frames out of 916 frames)	**16.5%** (151 frames out of 916 frames)

Table I. The results for the task: "Locating traffic signs". "Correct" answer means that EITHER the input frame has no traffic signs on it *and* there are no located areas on the output either, OR, there is at least one sign on the input *and* there are located areas on the output as well. The test video set included 916 real-life frames from different locations and with different lighting conditions.

Since Table I does not indicate the *accuracy* of the located areas ("ROIs"), we provide another table (Table II) showing these results.

	Correctly identified locations	Incorrectly identified locations
Altogether 490 located areas	**73.7%** (361 ROIs out of 490)	**26.3%** (129ROIs out of 490)

Table II: the accuracy of the identified locations. Only those frames are included, where there was at least one located area. A ROI is "correct" if there is a traffic sign at that very location, and "incorrect" otherwise. Thus, an answer belonging to one single frame can contain both correct and incorrect locations (see for example figure 5 d)

2.3.3.) Finding light sources

A trivial matter for people with normal vision but often a hard task for the blind ones, is to detect whether the lamps are switched on or switched off - for example after guests. According to our consultant from the "Hungarian National Association of Blind and Visually Impaired People" – with whom the tasks has been defined together – an algorithm solving this task could prevent much annoyance.

Here, the most important criterion is that the solution has to be independent of the input's actual brightness, that is, its' reliability should be the same in the case of a sun-drenched room and a dark cell.

The solution for this subtask differs from the former one in the sense that here we rely merely on retina channel information – instead of saliency maps. One channel proved to be enough for this task, namely the "Polar" channel, which seems to respond on light sources [2, 6] (figure 7). It gives strong reaction on primary light sources, both for natural (sun) and artificial ones (lamps) – and, to reflecting surfaces as well (mirrors, glass-tables, etc.) which cause a small error rate. Still, the accuracy this channel enables is very high: the ratio of the correct answers reaches 98-99% (see table III).

Since this channel responds on natural light sources as well (figure 7 c), the user is supposed to know where the window is, but this is not a real restriction in every-day practice. Otherwise, precise knowledge about the location of the lamp(s) is not a demand, since the visual environment can be scanned.

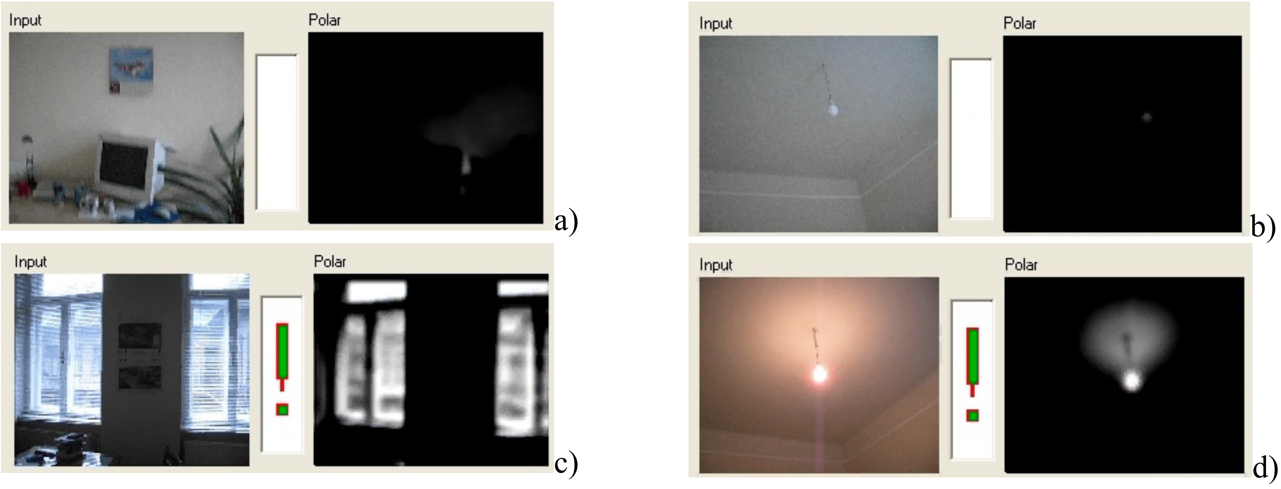

Figure 7: The "Polar" channel responds on light sources, both for natural ***(c)*** and for artificial ones ***(d)***. The pictures are taken from the test video set. All the four figures show the input on the left, and the corresponding output of the Polar channel, on the right. ***(a)*** and ***(b)***: a part of a bright room in day light; the Polar channel is basically silent. ***(d)*** had been recorded a few seconds after ***(b)***: the lamp is switched on, the Polar channel is excited. (The exclamation mark between the two channels indicates that the answer is: "there is light source on the input".) The Polar channel enables very high reliability for this task (see table).

According to the experiments, the Polar channel saturates (gives maximal response) on those areas where primary light sources are present, and give no answer elsewhere (figure 7). It follows that the accuracy of the algorithm is completely independent of the quality of the input video-flow, and also, it does not depend on the *brightness* either. The results depicted in table III are based on test videos made in sunshiny rooms.

	Correct answer	False answer
There *is* light source on the input video frame	98.8%	1.2%
There is *no* light source on the input video frame	99.38%	0.62%

Table III: The test results of the algorithm aiming to detect primary light sources, independently from the brightness of the input, or in other words, from the intensity values. The process is based on one of the mammalian retina channels (namely the "Polar" channel [2]), which reacts on light sources. The small error is due to reflecting surfaces (a glass table in our case). These values are based on the evaluation of test videos made on shiny rooms, including 1563 frames together.

2.3.4) Locating LED indicators

In many public buildings, offices and transport vehicles basic information is transmitted by LED indicators. The aim of this method again is to carry out a fast solution that localizes the areas that contain the indicators in question with high probability, on various indoor and outdoor real-life scenes. The main difficulty – except the bad resolution and the instability – originates from the *variety* of the possible inputs.

The test video set we have used includes multifarious scenes including different public and private places. Some of these can be seen on figure 8 and 9. On figure 8 we have also visualized those three channels that we have used for solving this task. These are the two color-opposition channels (blue-yellow and red-green) and the Delta channel. The function of the Delta channel has not yet been precisely formulated up to present (see above), but according to the observations, it gives significant response for small or fragile light sources as well (similarly to strong light sources).

From here, the selection algorithm is the following: if on a given location at least one of the two color opposition channels gave bigger response than a certain threshold, then a "fitness – value" would be calculated, being directly proportional to the three channel-data at the given point. Afterwards these values would be arranged into descending order, and the first few locations would be the *solution* for the given frame, that is *regions* that the algorithm defines as presumptive LED locations.

Table IV shows the results we have measured on this task. The results are based on the evaluation of 1207 frames. We have tested our method on real-life scenes, taken from different areas with various lighting conditions, reflecting areas, light sources, colors, etc. As it turned out, the algorithm is *not* sensitive to the quality of the input (e.g. resolution), to the lighting conditions or colors, either to the reflecting areas, but it *is* sensitive to colored lamps – which is not surprising since LEDs basically *are* small colored lamps, until no further object or pattern recognition algorithm is used.

	Correct answer	*False answer*
There *is* LED indicator on the input video frame (total 951 frames)	**96.6%** (919 frames out of 951 frames)	**3.4%** (32 frames out of 951 frames)
There is *no* LED indicator on the input video frame (total 256 frames)	**41%** (105 frames out of 256 frames)	**59%** (151 frames out of 256 frames)
Total (1207 frames)	**84.83%** (1024 frames out of 1207 frames)	**15.17%** (183 frames out of 1207 frames)

Table IV. The results for the task: "finding LED indicators". The values are based on the evaluation of 1207 frames. First row first column is the correct positive (96,6%), second row first column is the correct negative result (41%). The test database included complex real-life scenes with different lighting conditions, colored and reflecting areas and colored lamps. As it turned out, the algorithm in *not* sensitive to the quality of the input (resolution), to the lighting conditions and colors, either to the reflecting areas, but it *is* sensitive to colored

lamps – which the few frames (total 256) that did not contain LED happened to teemed in. The bad results are due to these lamps (figure 9 b). In the third row (“Total”), all the frames are counted, that is, “correct answer” indicates the percentage of the frames where either the input included LED indicator (one or more) and the output was at least one located area, or the input did not include LED indicator and the output had no located areas. Accordingly, the line “False” indicates the rest. The percentage means *frame* percentage.

Since the input frame may contain more than one LED indicator, and also, the output can be more than one located region (see figure 8 and 9), the evaluation of this task – similarly to the task of finding traffic signs – is not as straightforward as in the previous task, where the answer was binary (“there *IS* light source on the input”/“there is *NO* light source on the input”). Thus we give another table as well, which indicates the correctness of the locations which the algorithm has given as solutions. In contrast with table IV, table V depicts *ROI percentage* instead if frame percentage, that is, the ratio of the correct and false located areas. Only those frames are included, where there was at least one located area.

	Correctly identified locations	Incorrectly identified locations
Altogether 2075 located areas	**81.36%** (1688 ROIs out of 2075)	**18.64%** (387 ROIs out of 2075)

Table V: the accuracy of the identified locations. Only those frames are included, where there was at least one located area. A ROI is “correct” if there were a LED indicator at that very location, and “incorrect” otherwise. Thus, an answer belonging to one single frame can contain both correct and incorrect locations (see for example figure 9 b, where the marking of the colored lamp is incorrect (left hand side, top of the picture), while the sign on the elevator panel is correct – right hand side, top of the picture).

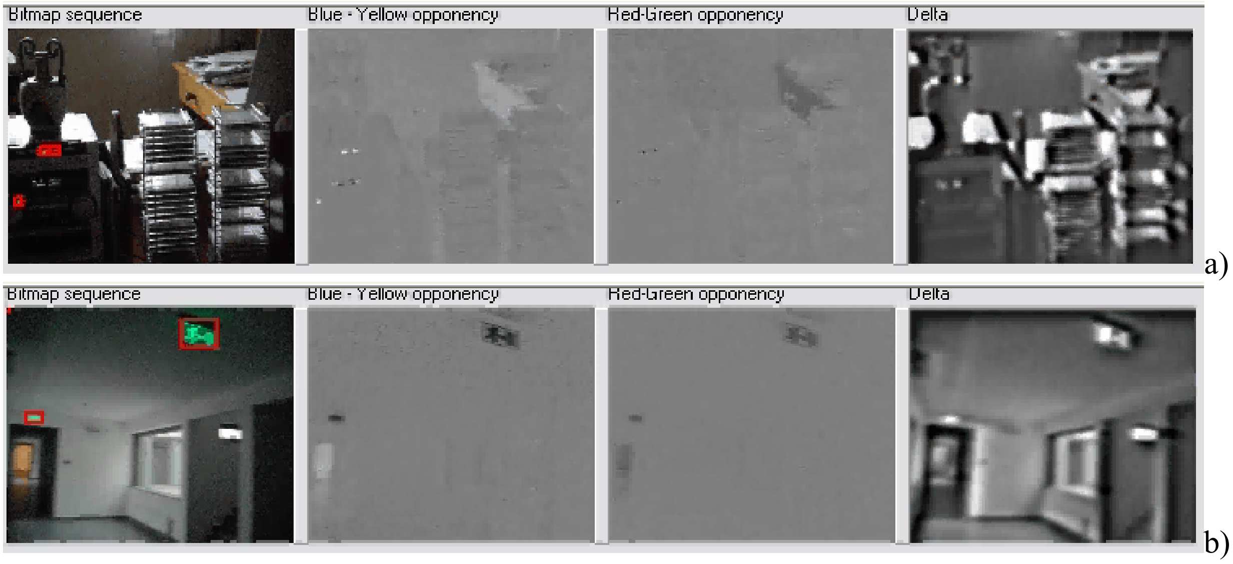

Figure 8. Two frames of the test database for the task “finding LED indicators”. The left-most pictures in both lines show the input with the identified locations on them. The other three pictures belong to those channels, whose data has been used in the execution of the task. These are the red-green and blue-yellow color opposition channels and the Delta channel. ***(a)*** LEDs belonging to a hi-fi set in a room. The various reflecting surfaces do not confuse the algorithm. ***(b)*** corridors in the university.

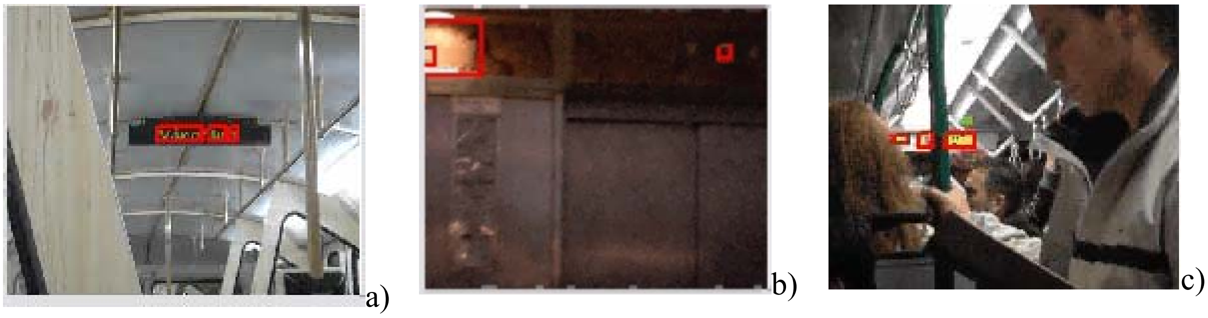

Figure 9. Some frames from the videos that we have used for testing the algorithm that finds LED indicators. ***a)*** rack-railway from the inside (these indicators show the name of the next stop and the actual time) ***b)*** a colored decorating lamp (left) and a LED indicator (right) showing the floor-number on a lift-panel in a department store. ***c)*** tram interior.

4) Future tasks

- During walking, a camera held in a hand, makes a quasi-periodic motion. Most of the people have their own way of "swinging" the phone, thus the transformation-parameters (vertical/horizontal shifts, the angle of the rotation, etc.) characterizes the certain users. These quasi-periodic parameter values could be learned during a certain amount of frames (and could even be adjusted during the entire usage), thus they become predictable for a given user. In this manner, by taking the predicted transformation values into account, the quality of the stabilization can be improved.
- The model described in this paper is *attentional* in the sense that it locates regions on the input where something important appears. Naturally arises the possibility of applying a more elaborated pattern or object recognition algorithm onto the selected area.
- Many possibilities lie in the retina channel decomposition. Thus, the further investigation of the individual channels can lead to a promising basis for different scene analyzer and object recognition algorithms. For example, some time-dependent channel (primarily the Transient, Beta and the Bistratified channels) seem to play an important role in separating the different objects from each other – although, this area needs further investigations.
- In a more elaborated version, the threshold values can be adjusted by a learning algorithm, and also could be adaptive according to the different scenes and tasks.

Acknowledgements

This work has been supported by the National R&D Agency of Hungary (HKTH) via the RET initiative, by the National Research Found (OTKA) and the Faculty of Information Technology of the Pazmany University.

References

[1] Roska, T., Karacs, K., Wagner, R., Lazar, A., Balya, D., Szuhaj, M.: Bionic Eyeglass: an audio guide for visually impared. Proceedings of the 1st Biomedical Circuits and Systems Conference, London, pp. 190-193 (2006)

[2] Balya, D., Roska, B., Roska. T.and Werblin, F. S.: A CNN framework for Modeling Parallel Processing in a Mammalian Retina. Int. J. on Circ. Theory and Appl. **30**, 363-393 (2002)

[3] Zitova, B ., Flusser, J.: Image registration: a survey. Image and Vision Comp. **21**, 977-1000 (2003)

[4] Otte, M., Nagel, H. H.:. Optical flow estimation: Advances and comparisons, Computer Vision, ECCV-94, **800**, 51-60, (1994)

[5] Horn, B., Schunck B.: Determining optical flow. Artificial Intelligence, **17**, 185-204, (1981)

[6] Roska B., Werblin, F. S.: Vertical interactions across ten parallel, stacked representations in the mammalian retina, Nature, **410**, 583-587 (2001)

[7] Kandel, E. R., Schwartz, J. H. and Thomas, M. J.: Principles of Neural Science, McGraw-Hill / Appleton&Lange, 3rd edition, New York (1996)

[8] Masland, R. H.: The fundamental plan of the retina, Nat. Neuroscience, **4**, 877-886 (2001)

[9] Chua, L. O., Roska, T.: Cellular Neural Networks and Visual Computing, Cambridge University Press, Cambridge (2002)

[10] Werblin, F. S., Roska, T., Chua, L. O.: The analogic cellular neural network as a bionic eye, Intl. J. of Circuit Theory and Applications, **23**, 541-569 (1995)

[11] Lazar, A. K., Wagner, R., Balya, D., Roska, T.: Functional representations of retina channels via the RefineC retina simulator, Paper presented at the 8th IEEE International Biannual Workshop on Cellular Neural Networks and their Applications, Budapest (2004)

[12] Itti, L., Koch, Ch: Computational modelling of visual attention. Nat. Neuroscience. **2**, 1-10 (2001)

[13] Barron, J.L., Fleet, D.J., Beauchemin, S.: Performance of optical flow techniques. International Journal of Computer Vision, **12(1)** 43-77 (1994)

Appendix

The normal flow algorithm

To estimate the instantaneous velocity field we model the motion image by a continuous variation of image intensity as a function of position and time. The intensity value on position (*x, y*) at time *t* is described by the *f(x, y, t)* intensity function.
If we expand this function in a Taylor series we get:

$$f(x+dx, y+dy, t+dt) = f(x,y,t) + \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial t}dt + HT \qquad \text{(A1)}$$

where 'HT' is for higher-order terms, which are usually ignored.
The crucial observation that is exploited, is that if the image at some time *t+dt* is a result of the original image at time *t* being moved translationally by *dx* and *dy*, then

$$f(x+dx, y+dy, t+dt) = f(x,y,t) \qquad \text{(A2)}$$

Thus, from equations (A1) and (A2) we get:

$$0 = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial t}dt, \quad \text{or, in other form:} \quad -\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} \qquad \text{(A3)}$$

$\frac{\partial f}{\partial t}$, $\frac{\partial f}{\partial x}$, and $\frac{\partial f}{\partial y}$ are measurable quantities, while $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are the quested values, namely the velocity in *x* and *y* directions.

Using the $\frac{dx}{dt} = u$ and $\frac{dy}{dt} = v$ notation, we get $-\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}u + \frac{\partial f}{\partial y}v$, (A4)

or, equivantly, $-\frac{\partial f}{\partial t} = \nabla f \cdot u$,

where ∇f is the spatial gradient of the image and ***u****=(u,v)* is the velocity vector.

The calculation steps

- *Measured values*: $I_x(=\frac{\partial f}{\partial x})$, $I_y(=\frac{\partial f}{\partial y})$, and $I_t(=\frac{\partial f}{\partial t})$, the intensity gradients

- *Calculated values (the estimations)*: $u(=\frac{dx}{dt})$ and $v(=\frac{dy}{dt})$

With these notations (A4) will be, for every pixel: $I_x u + I_y v + I_t = 0$ (A5),

The measured values:

The I_x, I_y *spatial gradients* can be determined with a convolution, where the kernel is the

[-1 8 0 -8 1] / 12 vector, which is a commonly used estimation in the literature [13]. (This kernel is applyed on the Gauss-filtered image, that is, we use the $\begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}$ as "A" template, accoring to the CNN terminology.)

The I_t *time gradient* is simply the difference between the two Gauss-filtered images.
(As it follows from the above process, these values are defined for each and every pixels, so I_x is not a scalar, but a matrix, and I_y, I_t similar.)

Defining the quested paramteres:

For *mapping function*, we choose a linear affine transformation, which can handle shifts in *x* and *y* directions, scaling, rotation and shear, as next:

$$u = a_0 + a_1 x + a_2 y \qquad \text{(A6)}$$
$$v = b_0 + b_1 x + b_2 y$$

where a_0, a_1, a_2, b_0, b_1 and b_2 are the parameters of the transformation, which we want to determine. In the matrix-form of (A6), the meaning of these parameters can be followed better:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ b_1 & b_b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad \text{(A7)}$$

In this equation $\begin{bmatrix} a_0 \\ b_0 \end{bmatrix}$ defines the translation in *x* and *y* directions, while $\begin{bmatrix} a_1 & a_2 \\ b_1 & b_b \end{bmatrix}$ describes the scaling, rotation and shear.

Thus, from (A5) and (A6), for every pixel we get:

$$(a_0 + a_1 x + a_2 y) I_x + (b_0 + b_1 x + b_2 y) I_y + I_t = 0 \qquad \text{(A8)}$$

which will be

$$\begin{bmatrix} I_x^{(1)} & I_x x^{(1)} & I_x y^{(1)} & I_y^{(1)} & I_y x^{(1)} & I_y y^{(1)} \\ I_x^{(2)} & I_x x^{(2)} & I_x y^{(2)} & I_y^{(2)} & I_y x^{(2)} & I_y y^{(2)} \\ & & . & . & . & \\ I_x^{(k)} & I_x x^{(k)} & I_x y^{(k)} & I_y^{(k)} & I_y x^{(k)} & I_y y^{(k)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix} = - \begin{bmatrix} I_t^{(1)} \\ I_t^{(2)} \\ \ldots \\ I_t^{(k)} \end{bmatrix} \qquad \text{(A9)}$$

where *k* (the number of rows) is the number of the pixels.

With the notations:

$$A = \begin{bmatrix} {I_x}^{(1)} & I_x x^{(1)} & I_x y^{(1)} & {I_y}^{(1)} & I_y x^{(1)} & I_y y^{(1)} \\ {I_x}^{(2)} & I_x x^{(2)} & I_x y^{(2)} & {I_y}^{(2)} & I_y x^{(2)} & I_y y^{(2)} \\ & & . & . & . & \\ {I_x}^{(k)} & I_x x^{(k)} & I_x y^{(k)} & {I_y}^{(k)} & I_y x^{(k)} & I_y y^{(k)} \end{bmatrix}$$, and $$P = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix}$$, we get

$A \cdot P = -I_t$, from where $P = A^+ \cdot (-I_t)$ (A10)

where ‘+’ denotes pseudo-inverse, and the *P* vector containes the transformation-parmeters, which we were looking for.