

# 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:

$$\mathbf{v}^*(\mathbf{x}) = \operatorname{argmin}_{\mathbf{v}(\mathbf{x})} \sum_{\theta \in O(\mathbf{x})} \left( \|\mathbf{v}_{c,\theta}(\mathbf{x})\| - \mathbf{v}(\mathbf{x})^\top \frac{\mathbf{v}_{c,\theta}(\mathbf{x})}{\|\mathbf{v}_{c,\theta}(\mathbf{x})\|} \right)^2, \quad (6)$$

where  $O(\mathbf{x})$  is the set of orientations at which valid component velocities have been obtained for pixel  $\mathbf{x}$ .

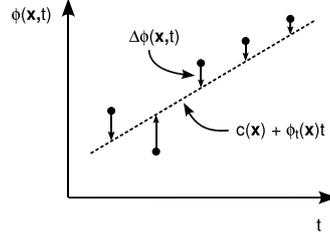


Figure 1: Temporal phase gradient linearization. For each orientation and pixel, the temporal phase gradient  $\phi_t(\mathbf{x})$  is computed by fitting a line through the spatial phases  $\phi(\mathbf{x}, t)$  computed at each frame. The proposed stabilization method aims at minimizing the deviations  $\Delta\phi(\mathbf{x}, t)$  from this estimated line by applying a global 3D stabilizing rotation  $\Delta\omega(t)$  to each frame.

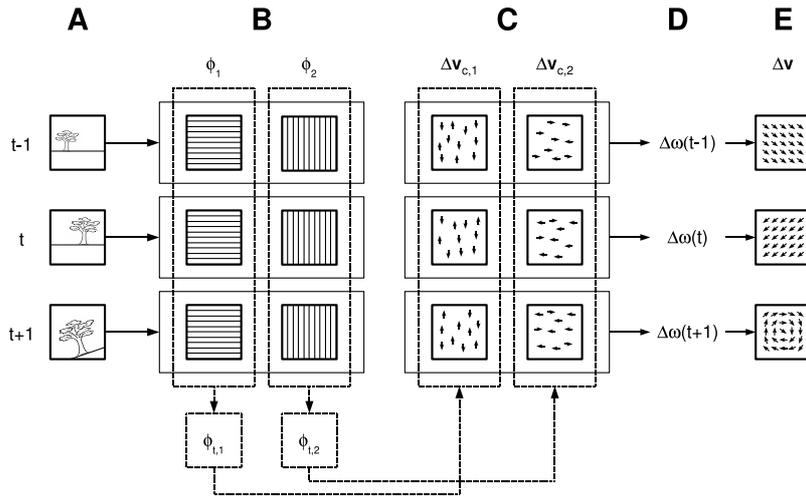


Figure 2: Stabilization overview. **(A)** A sliding window (consisting of three frames in the figure) is used to compute optic flow for the central frame  $t$ . **(B)** The spatial phase  $\phi_\theta$  is computed for each pixel, orientation  $\theta$  (two orientations in the figure) and frame  $t$ . 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:

$$\Delta \boldsymbol{\omega}^*(t) = \operatorname{argmin}_{\Delta \boldsymbol{\omega}(t)} \sum_{\mathbf{x}, \theta} \left[ \|\Delta \mathbf{v}_{c,\theta}(\mathbf{x}, t)\| - (B(\mathbf{x})\Delta \boldsymbol{\omega}(t))^T \frac{\Delta \mathbf{v}_{c,\theta}(\mathbf{x}, t)}{\|\Delta \mathbf{v}_{c,\theta}(\mathbf{x}, t)\|} \right]^2. \quad (11)$$

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.

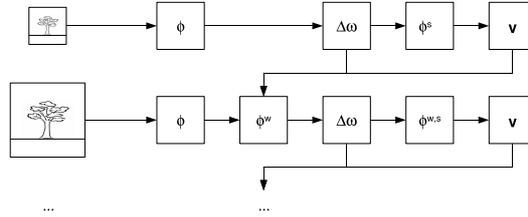


Figure 3: Multiscale stabilization.

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.

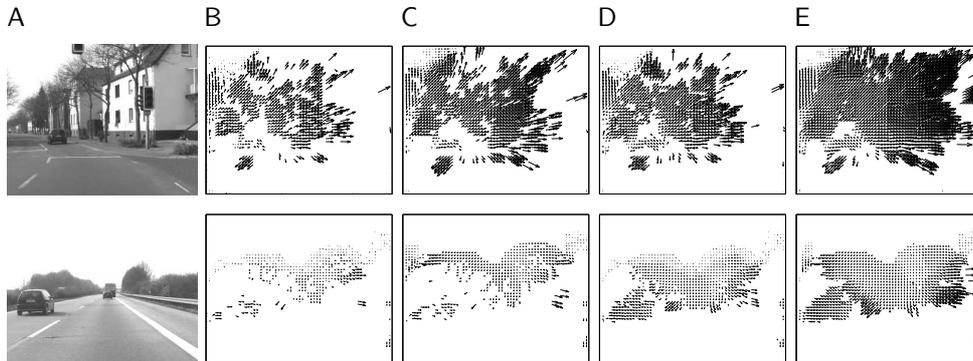


Figure 4: Example images (A) and flow fields (B–E) obtained on the *city* (top row) and *mway* (bottom row) sequence without stabilization using (B) single scale and (C) multiscale optic flow, and with the proposed stabilization using (D) single scale and (E) multiscale optic flow. All flow fields have been subsampled and scaled 10 times.

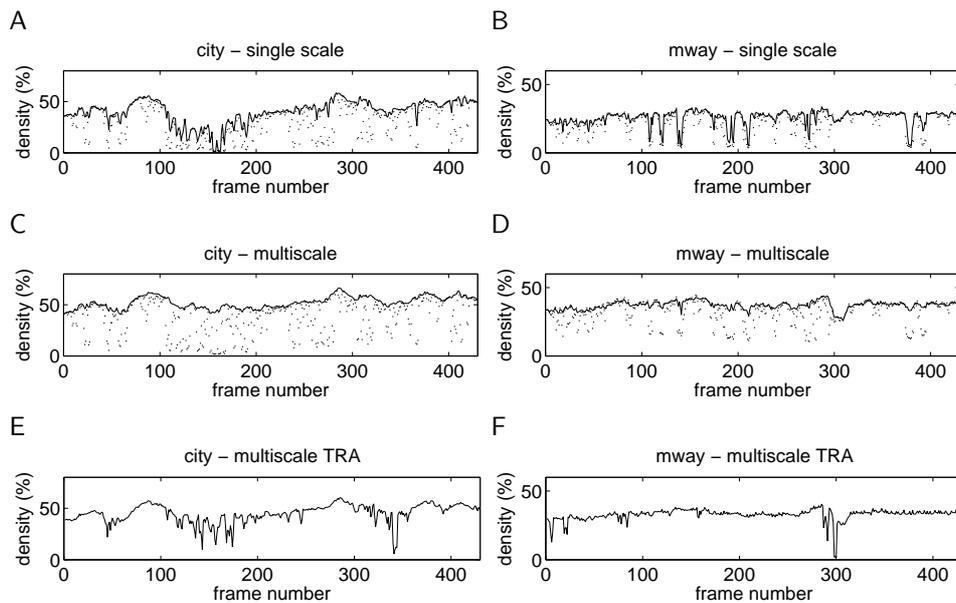


Figure 5: Optic flow field density. (A–D) Results obtained without stabilization (black dots) and after stabilization with the proposed method (solid line) over the entire *city* (left) and *mway* (right) sequences. The first and second row correspond to the results obtained with the single and multiscale 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. Feeman 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.