International Journal of Computer Vision

DOI: 10.1007/s11263-006-8893-6

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; Taalebinezhaad, 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 (Taalebinezhaad, 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

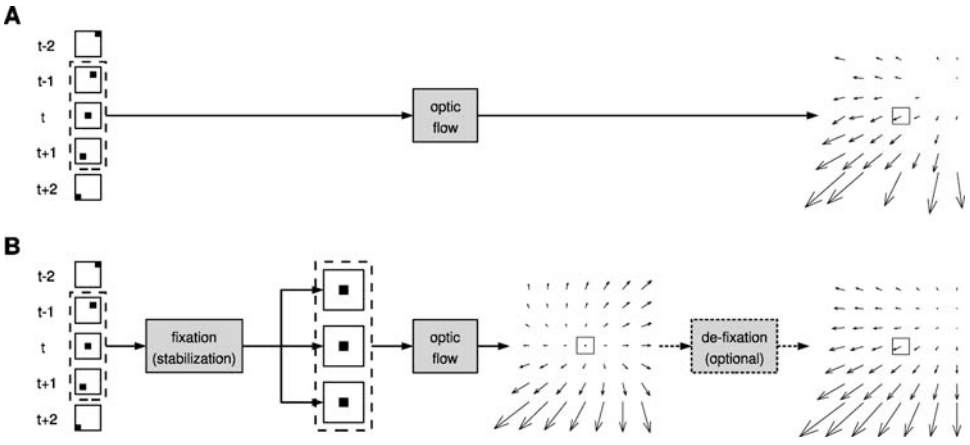

Figure 1. Classical optic flow computation **(A)** and the proposed method **(B)**. The dashed box marks the sliding temporal window used in computing the optic flow at time t. Without stabilization, the flow field is sparse and noisy (right flow field in **A**). The small filled squares mark the location 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[1]. 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:

$$\dot{x} = \omega_x \frac{xy}{f} - \omega_y \left(f + \frac{x^2}{f} \right) + \omega_z y \qquad (1)$$

$$\dot{y} = \omega_x \left(f + \frac{y^2}{f} \right) - \omega_y \frac{xy}{f} - \omega_z x \ , \qquad (2)$$

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):

$$\dot{x}_1 = -x_1 \qquad (3)$$

$$\dot{y}_1 = -y_1 \ . \qquad (4)$$

Since we only consider x- and y-axis rotations in the stabilization, a unique compensatory 3D rotation

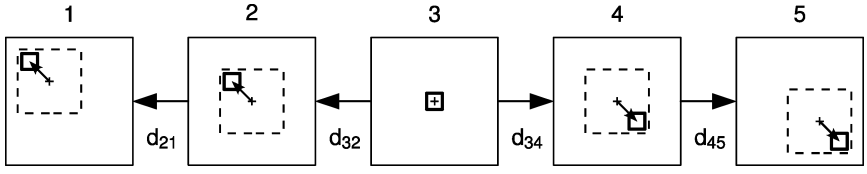

Figure 2. Stabilization by means of fixation. The central image region of frame 3 is backward and forward tracked to frames 1 and 5 respectively. In this way, the individual displacements $\mathbf{d}_{ij}$, denoting the movement of the feature from frame i to frame j, are determined.

satisfies this requirement:

$$(\omega_x, \omega_y, \omega_z) = (-\frac{y_1 f}{f^2 + x_1^2 + y_1^2}\ ,\ \frac{x_1 f}{f^2 + x_1^2 + y_1^2}\ ,\ 0)\,. \tag{5}$$

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 Ω that yields the correct (pixelwise) displacement estimates:

$$\sum_{\mathbf{x}\in\Omega} \Big(I_x(\mathbf{x}, t)s_x + I_y(\mathbf{x}, t)s_y + I_t(\mathbf{x}, t) \Big)^2 , \qquad (6)$$

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

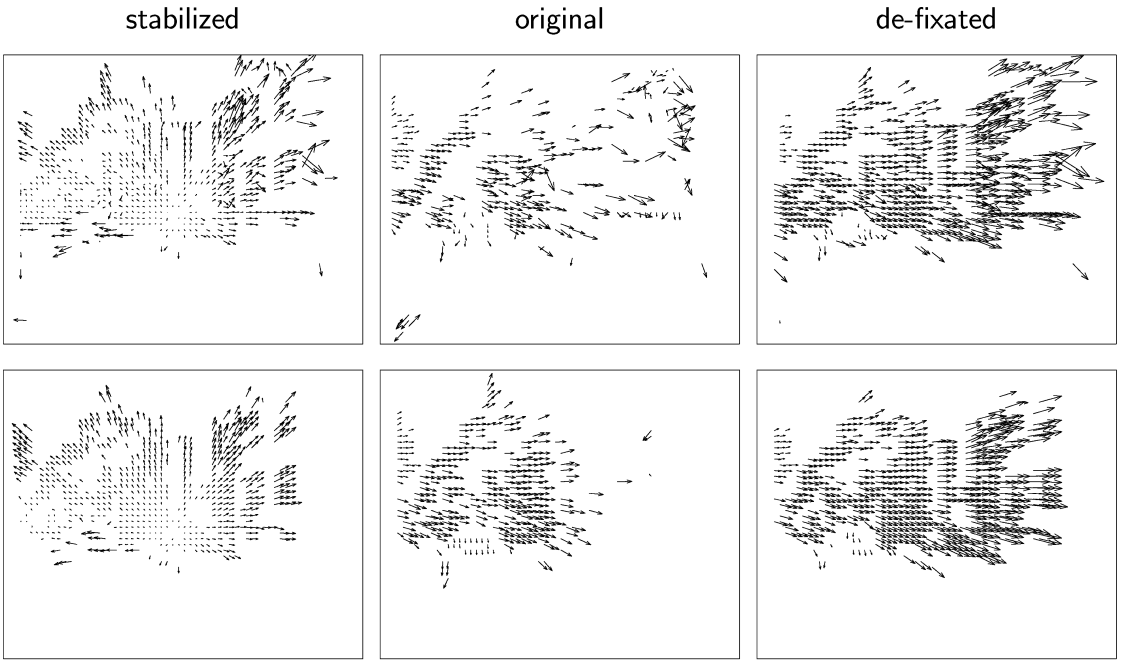

Figure 3. Flow fields computed for the *city3* frame shown in Figure 4. The flow has been computed using LUC (top row) and GAU (bottom row). The left and middle columns contain the flow fields computed respectively with and without stabilization. The right column contains the stabilized flow fields after 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[2]. All sequences are 18 seconds long and contain 450 frames at a resolution of 638 × 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 15×15 pixels and cubic interpolation is used to achieve subpixel 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.

	LUC				GAU			
seq	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.

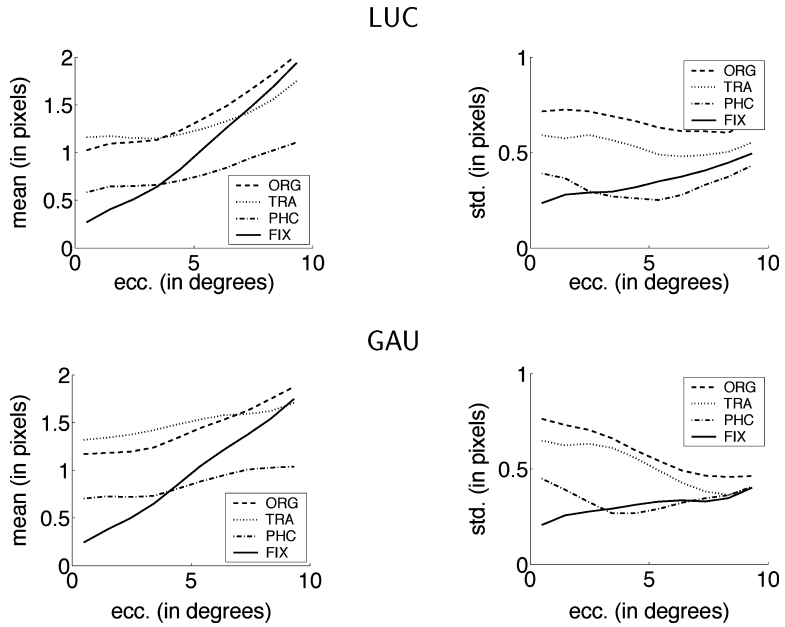

Figure 5. Mean (left column) and standard deviation (right column) of the optic flow vector magnitude as a function of eccentricity with and without stabilization. The results have been summarized over all sequences and are shown in the top and bottom row for LUC and GAU respectively.

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 Drivsco.

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.

Taalebinezhaad, 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.