A Local Algorithm for the Computation of Optic Flow via Constructive Interference of Global Fourier Components

Babette Dellen[1,2] and Florentin Wörgötter[3]
[1] BCCN Göttingen, MPI for Dynamics and Self-Organization, Germany
[3] BCCN Göttingen, University Göttingen, Germany
bkdellen@bccn-goettingen.de

Abstract

A novel Fourier-based technique for the estimation of optic-flow fields from image sequences is proposed. In this method, the instantaneous velocities of local image points are inferred directly from the global 3D Fourier components of the image sequence. This is done by selecting those velocities for which the superposition of the corresponding Fourier gratings leads to constructive interference at the image point. Hence, uncertainties caused through a windowed measurement process typical for local methods do not arise. The algorithm is tested on both synthetic and real image sequences and the results are compared to those of local techniques for optic-flow computation.

1 Introduction

The computation of velocity is intrinsically connected to the representation of images in Fourier space. All the nonzero power associated with a translating 2D-pattern lies on a plane through the origin in Fourier space. The pattern velocity determines the orientation of the plane, and can be derived from the nonzero Fourier components by finding the velocity for which the constraint lines of the Fourier components intersect. However, usually the image sequence contains more than one object. Most optic-flow techniques try to resolve this problem by restricting their computations to a small spatiotemporal region of the image sequence [3, 6]. Given that this region is chosen to be sufficiently small, it contains, in the ideal case, only the object of interest, which will then have an approximately constant velocity. There is a catch though: Since the measurement of frequencies is based on spatiotemporal sampling, the velocity estimates become less accurate with decreasing window size. This phenomenon is known as the uncertainty principle in signal processing [12]. As a consequence, local methods [3, 6] can only compute accurate velocity estimates at points which have sufficient image structure, such as edges.

Several approaches have been proposed to facilitate image segmentation by representing images in the global Fourier domain. Vernon developed a Fourier-based method for segmenting images which are composed of an occluding foreground and occluded background for situations in which the object velocity is constant and normal to the principal

[2] The work has received support from the EU Project Drivsco under Contract No. 016276-2

BMVC 2008 doi:10.5244/C.22.79

ray of the image sensor [11]. Dellen *et al.* demonstrated that objects which are moving in opposing directions are separable in Fourier space [5]. Global Fourier components have also been used to reduce motion estimation errors in a hybrid approach [4]. However, a solid brigde between spatiotemporal localization and velocity measurements of image points is still missing. To fill this gap, we propose a novel algorithm for the estimation of optic flow which derives velocity estimates from the global Fourier components of the image sequence by reconstructing spatiotemporal position through constructive interference. Using this method, local velocity estimates can be assigned to image parts for which other local methods fail due to the constraints imposed by local windowing.

2 Algorithmic framework

The visual scene can be represented as a three-dimensional discrete function of intensity values $I(\mathbf{x},t)$, where $\mathbf{x}=(x,y)$ defines the spatial dimensions and t the temporal dimension. By means of Fourier decomposition, the image sequence can be described as a superposition of translating gratings, such that

$$I(\mathbf{x},t)=\sum_{\mathbf{k},k_t} A(\mathbf{k},k_t)\cos(\mathbf{k}\cdot\mathbf{x}-k_t t)+B(\mathbf{k},k_t)\sin(\mathbf{k}\cdot\mathbf{x}-k_t t) \quad , \tag{1}$$

where $\mathbf{k}=(k_x,k_y)$ is a wave vector with spatial frequencies k_x and k_y, and k_t is a temporal frequency. The amplitudes $A(\mathbf{k},k_t)$ and $B(\mathbf{k},k_t)$ depend on the spatial and temporal frequencies of the gratings and represent the image sequence in 3D Fourier space. Each grating moves with a velocity $\mathbf{v}=k_t\mathbf{k}/k^2$, where $k=|\mathbf{k}|$ is the absolute spatial frequency of the grating. The combined movements of the gratings contributing to a particular point $(\mathbf{x},t)$ of the image sequence determine the local velocity $\mathbf{V}(\mathbf{x},t)$ at this point. For illustration of the principle of constructive interference, see Fig. 1A. Importantly, for non-transparent conditions, only intensities belonging to a single object can be represented at a local point $(\mathbf{x},t)$. The contribution of a grating to the velocity of a point $(\mathbf{x},t)$ of the image sequence can be quantified by assigning a weight to every grating. For example, a grating which has a negative amplitude at a point of positive intensity is contributing destructively, while a grating of positive amplitude contributes constructively. Thus, weights depend not only on the amplitude of the grating at this point, but also on the intensity of the point itself, such that

$$w(\mathbf{x},t,\mathbf{k},k_t)=[A(\mathbf{k},k_t)\cos(\mathbf{k}\cdot\mathbf{x}-k_t t)+B(\mathbf{k},k_t)\sin(\mathbf{k}\cdot\mathbf{x}-k_t t)]\,\mathrm{sign}[I(\mathbf{x},t)] \quad , \tag{2}$$

where $\mathrm{sign}[\pm a]=\pm 1$ is the sign function. Hence, for each point $(\mathbf{x},t)$ of the image sequence, we obtain a set of weighted velocities $\{\mathbf{v}(\mathbf{k},k_t);w(\mathbf{x},t,\mathbf{k},k_t)\}$. The set of weighted velocities can be used to estimate the local velocity $\mathbf{V}(\mathbf{x},t)$ of the image point since all gratings belonging to an object moving with a velocity $\mathbf{v}$ fulfill the motion-constraint equation $k_t=\mathbf{v}\cdot\mathbf{k}$ [1].

We estimate the local velocity by employing the following voting scheme: Each component velocity votes with its assigned weight for all local velocities which lie along the corresponding constraint line, yielding a map of votes

$$m(\mathbf{U},\mathbf{x},t)=\sum_{\mathbf{k},k_t} w(\mathbf{x},t,\mathbf{k},k_t,\mathbf{x},t)\exp[-(k_t-\mathbf{U}\cdot\mathbf{k})^2/(\xi\,|\mathbf{k}|)^2] \quad , \tag{3}$$

Sequence	ρ^{DW}	$\mathrm{E}_{\mathrm{a}}^{\mathrm{DW}}$	$\rho^{\mathrm{DW,s}}$	$\mathrm{E}_{\mathrm{a}}^{\mathrm{DW,s}}$	ρ^{LK}	$\mathrm{E}_{\mathrm{a}}^{\mathrm{LK}}$	ρ^{GH}	$\mathrm{E}_{\mathrm{a}}^{\mathrm{GH}}$
Trans. Square	0.041	2	-	-	0.031	35	0.041	15
Trans. Tree	0.1	3.5	1	0.59	0.398*	0.66*	0.678*	2.67*
Div. Tree	0.044	4.37	1	1.93	0.482*	1.94*	0.773*	4.07*
Yosemite	0.018	6.29	0.74	4.9	0.351*	4.1*	0.347*	4.4*
Rubber Whale	0.012	5.49	1	11.38	1	11.16	0.598	10.14
Hydrangea	0.014	12.52	1	9.68	1	15.86	0.366	6.86

Table 1: Average angular error E_{a} and density ρ of our algorithm (DW) for various image sequences. Subsequent smoothing of the optic-flow field decreases in many cases the average angular error while increasing the density (DW,s). But the error at motion boundaries may increase drastically (Fig. 5A-F). The results of the algorithms of Lukas and Kanade (LK) and Gautama and van Hulle (GH) are shown for comparison. However, it is difficult to compare methods using different degrees of smoothing. For the Yosemite sequence, the area of the clouds was excluded from evaluation. * Values taken from [3,7].

where the parameter ξ is the width of the Gaussian. The Gaussian function, which reaches its maximum at $k_t = \mathbf{U} \cdot \mathbf{k}$, implements the motion-constraint equation. If ξ is large, gratings with a combination of frequencies in the neighborhood of the selected velocity also contribute to the voting. If ξ is small, almost only those gratings contribute to the voting which exactly fullfill the motion-constraint equation exactly. A large ξ can be advantageous if the motion of the image point is accelerated and thus smeared in Fourier space.

The map of votes peaks at $\mathbf{U} = \mathbf{V}(\mathbf{x},t)$ if the velocity signal is sufficiently strong, where $\mathbf{V}(\mathbf{x},t)$ is the local velocity of the point. Thus, we compute the velocity estimate $\mathbf{V_e}(\mathbf{x},t)$ by finding the maximum of the map of votes and taking its argument

$$\mathbf{V_e}(\mathbf{x},t) = \arg\{\max[m(\mathbf{U})]\} \quad . \tag{4}$$

An overview of the results of the algorithm is given in Table 1.

2.1 Confidence values

We measure the "confidence" $G(\mathbf{x},t)$ of the velocity estimated at $\mathbf{x}$ by computing the correlation coefficient of the map of votes with a Gaussian centered at $V_e(\mathbf{x},t)$

$$G(\mathbf{x},t) = C\left\{\exp\left[-(\mathbf{U}-\mathbf{V_e}(\mathbf{x},t))^2/\sigma^2\right], m(\mathbf{U},\mathbf{x},t)\right\} \quad , \tag{5}$$

where σ is a parameter determining the width of the Gaussian. The function G provides a confidence measure for each estimated local velocity value. For example, if the velocity signal is too weak, no peak will emerge from the map of votes, and Eq. 4 will return a wrong velocity estimate. In this case, the confidence will be small and the velocity estimate can be discarded. We use a parameter τ to threshold the confidence function.

2.2 Filtering

If an image point is part of a large homogeneous image area, the low spatial frequencies are usually static even if the respective object is moving due to undersampling. This error source can be reduced by preprocessing the image sequence with a spatiotemporal high-pass Butterworth filter

$$\Psi_f = 1/[1+\tau_f/(k_x^2+k_y^2+k_t^2)] \quad , \tag{6}$$

where τ_f is a threshold parameter.

3 Results

The algorithm is tested on several artificial and real image sequences. Based on the results of a sensitivity analyis (not shown), the parameters of the core algorithm are set to $\xi = 0.3$ and $\sigma = 0.6$ for all simulations. The parameter τ_f of the high-pass spatiotemporal filter is varied depending on the length of the image sequence. Short image sequences profit from a larger τ_f. For the computation of the map of votes, a velocity resolution of 0.1 pixel/frame was used. The absolute (end-point) error and the angular error are defined as in [3]. Computations were performed using MATLAB on a 1.73 GHz Intel Core Duo Processor. For a velocity range of -2 to 2 pixels/s, the programm required 187 s to compute the optic-flow fields of an image sequences containing 20 frames of size 100×100 pixels, corresponding to an approximate computation time of 9 s/frame.

3.1 Basic object motion

We illustrate the algorithm first on a sequence which contains a square, 10 pixels wide, which moves on a white background with a velocity of $\mathbf{v} = [1,1]$. The translating-square sequence contains 24 frames, which is about the number of frames movies display within a second (Fig. 1B, left panel). The global Fourier transform is taken over the whole sequence, as explained in Section 2, and the weights are computed for each point of frame 12 of the sequence. We compute the map of votes for velocities $\mathbf{U}$, where the component velocities in x and y direction are ranging from -3 to 3 pixels/frame in steps of 0.1 pixels/frames. The map of votes for an image point belonging to the square (labeled a) is presented in Fig. 1C. A clearly distinguishable peak is visible in the map of votes, with a maximum positioned at $\mathbf{U} = [1,1]$.

In contrast, point b in the surround of the square, which is a large homogenous image region, returns a flat map of votes, and no clear peak can be distinguished (Fig. 1D). This is reflected in the confidence $G(\mathbf{x})$ of the velocity estimate, presented in Fig. 1E. All points on the square have a high confidence, while points in the homogenous surround have not. Thresholding $G(\mathbf{x})$ with $\tau = 0.4$, returns only velocity estimates which have a high confidence. The x and y components of the estimated velocities are given in Fig. 1F-G, respectively. The algorithm returns estimates for points lying on the square. Remarkably, the estimated velocities of points belonging to the edge of the square are not afflicted with the aperture problem, since no windowing is performed. However, velocity estimates for points in the homogenous surround cannot be given. This merely demonstrates the

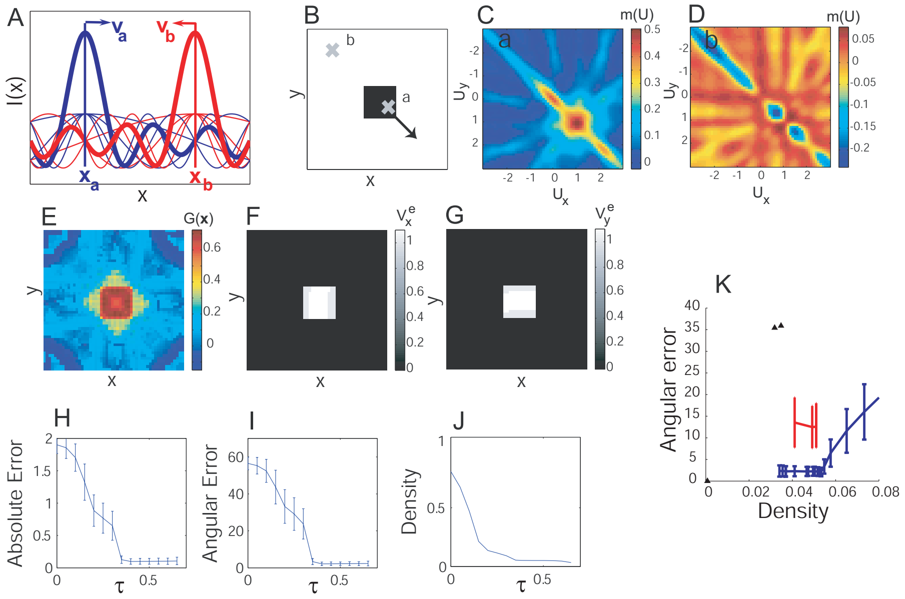

Figure 1: **A** The Fourier components of two objects (blue and red lines), moving with v_a and v_b, superpose constructively (bold blue and red lines) at the instantaneous location of the obects, x_a and x_b, respectively. **B** Schematic of translating square. **C** Map of votes for point a of the square, showing a clear peak. **D** Map of votes for point b in the homogeneous surround. **E** The confidence $G(\mathbf{x})$ as a function of position. **F-G** The estimated velocities in x and y direction. **H-J** Abolute error, angular error, and density as a function of the threshold τ. **K** Performance of optic-flow techniques for the translating-square sequence: Lucas and Kanade (black triangles), Gautama and Van Hulle (red), and our approach (blue).

ambiguity of the vision problem (correspondence problem), and is not a insufficiency of the algorithm, since the homogeneous background contains all possible velocities.

To quantify the performance of the algorithm, we compute the absolute error and the angular error as a function of of τ, together with the standard error (Fig. 1H-I, respectively). In Fig. 1J, bottom panel, the density of the sequence is plotted as function of τ. The error measures decrease with increasing threshold and settle at low values close to the resolution of the map of votes of 0.1 pixels/frame.

We compare the performance of the algorithm with the outcome of other optic-flow techniques, a phase-based approach by Gautama and Van Hulle [7] and a differential technique by Lucas and Kanade [9] (Fig. 1K). For our algorithm, we achieved decreasing densities by increasing the threshold τ. For the phase-based approach, we use parameters as in [7], namely $N_{min} = 7$ and τ_l ranging from 0.001 to 0.02 in steps of 0.002. The choice of the parameters however does not significantly alter the result of the algorithm. The angular error is above the angular error of our approach. The angular error of the Lucas and Kanade algorithm for a window size of 5×5 pixels^2 reaches a lower value than our algorithms, however, the density is very low. Different densities were obtained by thresholding the absolute gradient squared, leading to maximum confidence values at the corners of the square. At higher densities though, the performance of the Lucas and Kanade algorithm is less good than our approach. The large error at higher densities for the Lucas and Kanade algorithm is due to the aperture problem, which produces wrong velocities at the edge of the square. At very low densities, only the corners of the square are left, which are not afflicted with the aperture problem due to their higher intrinsic dimensionality [8]. The optic-flow algorithm by Gautama and Van Hulle produces good velocity directions at the edges because the method uses various spatial filters. The angular error is above our approach and below the angular error of Lucas and Kanade.

3.2 Translating and diverging tree sequence

The translating- and diverging-tree sequences are artificial sequences generated from a real image of a tree (Fig. 2A) [3]. In the translating-tree sequence, the camera moves to the right, while filming a picture of a tree, which is slanted with respect to the fronto-parallel plane. The velocities are parallel to the x-axis and range form 1.73 to 2.26 pixels/frame. In the diverging tree seequence, the camera recedes from the tree image, causing the optic-flow field to diverge from the center of expansion which is located in the middle of the image. The speed ranges from 1.29 to 1.86 pixels/frame.

We investigate the influence of high-pass spatiotemporal filtering with $\tau_f = 0.1$ on the performance of the algorithm. In Fig. 2B, the angular error is plotted as a function of the density of the estimated flow field for the raw and the high-pass spatial frequency filtered translating-tree sequence in black solid and black dotted lines, respectively. Decreasing densities are obtained by increasing the threshold τ. At equal densities, the angular error of the high-pass spatiotemporal frequency filtered sequence is below the angular error of the raw sequence. For the diverging-tree sequence, the performance of the high-pass spatiotemporal frequency filtered sequence is clearly improved compared to the raw sequence (Fig. 2C). The error statistics for the example of the translating and diverging tree sequence demonstrate that preprocessing can improve optic-flow estimation significantly. Angular errors have been measured for a variety of optic-flow techniques for the both the translating and diverging tree sequence. The angular errors of the different techniques range from $17.66 - 0.35$ deg and can be found in [3].

The estimated flow field for the raw translating-tree sequence and for the high-pass spatiotemporal frequency filtered sequence are shown for $\tau = 0.55$ are shown in Fig. 2D-E, respectively.

3.3 Real sequences

The algorithm is tested on three real sequences, i.e. the Hamburg-taxi sequence [3], the Rubber-Whale sequence, and Hydrangea (http://vision.middlebury.edu/flow/eval/)

In the Hamburg taxi sequence, a street scene is shown with four moving objects: (i) a taxi turning the corner (ii) a car in the lower left, driving from the left to the right (iii) a van in the lower right driving from the right to the left, and (iv) a person walking in upper left. Image speeds of the four moving objects are approximately 1.0, 3.0, 3.0, and 0.3 pixels/frame, respectively. A snapshot shown in Fig. 3A. The sequence contains 20 frames. The optic-flow field at frame 10 for $\tau = 0.55$ and $\tau_f = 0.05$ is shown in Fig. 3B. The optic-flow field is scaled by a factor 2 for reasons of better display. By means

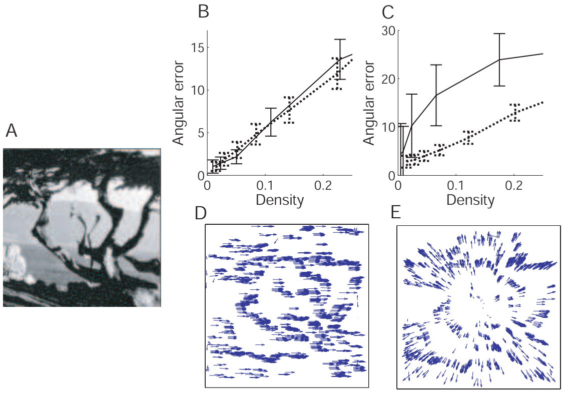

Figure 2: **A** Snapshot of the translating-tree sequence. **B** Translating-tree sequence: Angular error as a function of image density ρ for the raw sequence (black solid line) and the high-pass spatiotemporal frequency filtered sequence with $\tau_f = 0.1$ (black dotted line). **C** Diverging tree sequence: Angular error as a function of image density ρ for the raw sequence (black solid line) and the high-pass spatial frequency filtered sequence with $\tau_f = 0.1$ (black solid line). **D** Optic-flow field with $\tau = 0.55$ for the translating-tree sequence. **E** Optic-flow field with $\tau = 0.55$ for the diverging-tree sequence.

of manual segmentation, we computed the average velocity for each moving object: (i) $\bar{\mathbf{V}}_e = (-0.7,-0.7)$, (ii) $\bar{\mathbf{V}}_e = (2.7,0.5)$, (iii) $\bar{\mathbf{V}}_e = (-2.6,-0.3)$, and (iv) $\bar{\mathbf{V}}_e = (-0.3,0)$ pixels/frame.

We tested the algorithm on other real sequences from [3], i.e. the rubic cube, SRI-trees, NASA, and a scene showing a walking person, and obtained qualitatively overall similar results (not shown).

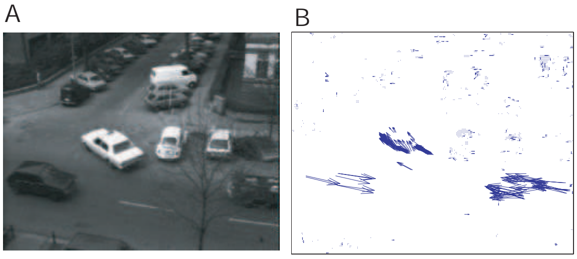

Figure 3: Hamburg taxi sequence **A** Frame 10 of the original sequence. **B** Estimated optic-flow field at frame 10 for $\tau = 0.55$ and $\tau_f = 0.05$.

The rubber-whale sequence from the Middlebury flow database contains eight frames and features complex motion discontinuities. Frame 4 of the sequence is shown in Fig. 4A.

The estimated color-coded optic-flow field for $\tau_f = 0.2$ and $\tau = 0$ is presented in Fig. 4B. Due to the local computations, the flow field is noisy, however, since no smoothing through local windowing is introduced, the motion boundaries are sharp. The angular error drops to ≈ 5 deg at low densities (see Fig. 4C).

We performed the same computations for the Hydrangea sequence, containing eight frames. The results are shown in Fig. 4D-F. For this sequence, the performance of the algorithm was less good than for the previous sequences with an angular error of ≈ 10 deg at low densities.

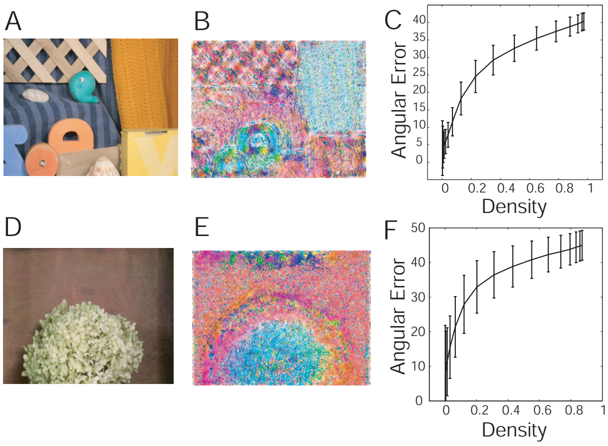

Figure 4: **A** Frame 4 of the Rubber-whale sequence. **B** Estimated optic-flow field at frame 10 for $\tau = 0$ and $\tau_f = 0.2$. **C** Angular error as a function of density. **D** Frame 4 of the Hydrangea sequence. **E** Estimated optic-flow field at frame 10 for $\tau = 0$ and $\tau_f = 0.2$. **F** Angular error as a function of density.

3.4 Smoothing

The effect of smoothing on the performance of the algorithm is investigated on the example of an artificial image sequence containing to random-dot patterns moving in opposite directions, one occupying the left side of the image, the other occupying the right side (Fig. 5A). The estimated color-coded optic-flow field $\tau_f = 0.05$ and $\tau = 0$ is presented in Fig. 5B. Motion boundaries are sharp, but the velocity estimates itself are noisy. We compute the absolute error of the optic-flow field and average along the y axis. The average absolute error is then plotted as function of position along the x axis (Fig.5C, red). The absolute error is noisy with an average of about 0.2 pixels/frame. The absolute error is drastically reduced if we only choose velocity estimates with confidences larger than $\tau = 0.5$ (Fig. 5C, black). We introduce a simple smoothing operation which averages over neighboring velocity estimates. The absolute error drops dramatically in homogenous velocity regions, however, at the motion boundary, the error increases (Fig. 5C, blue). Similar effects are observed for the Lucas and Kanade algorithm [9]. For a window-size of 1 pixel, the absolute error is large but does not change at the motion boundary (Fig. 5D, red). Note that the error of the unsmoothed Lucas and Kanade is much higher than the error of our algorithm. Choosing a larger window of 5 pixels, causes the error to drop in the homogenous region and to rise at the motion boundary (Fig. 5D, blue). We also computed the absolute error for two different smoothing parameters $\alpha = 0.05$ and $\alpha = 1$ of a global differential method [10] and observed the same behavior (Fig. 5E). This illustrates that smoothing reduces the error in certain image regions, however, at motion boundaries, the error increases.

We tested several smoothing operations on our algorithm (see Fig. 5F). For the translating and diverging-tree sequence, angular errors dropped below 2 deg. For the Yosemite sequence we obtained an average angular error of 4.9 deg (results not shown).

We further measured the absolute error as a function of the standard deviation of Gaussian noise added to the random-dot image sequence (see Fig. 5G). A noise level of 0.2 corresponds to 20% of the gray-value range of the image sequence.

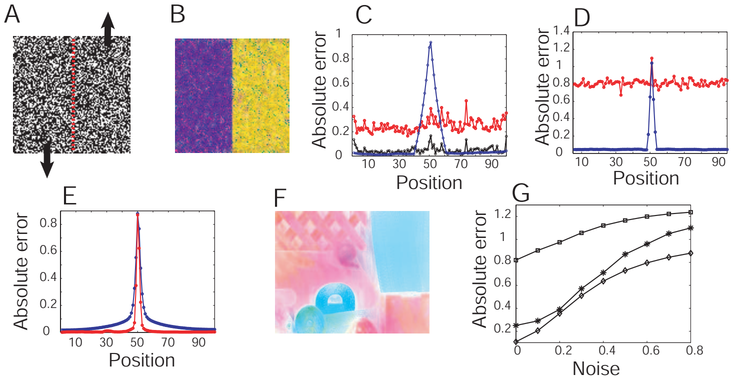

Figure 5: Effects of smoothing. **A** Schematic of the image sequence. **B** Optic-flow field for $\tau = 0$ and $\tau_f = 0.05$. **C** Absolute error as a function of position for $\tau = 0$ (red), $\tau = 0.5$ (black), and $\tau = 0$ with smoothing (blue). **D** Absolute error of the Lucas and Kanade algorithm as a function of position for window sizes of 1 pixel (red line) and 5 pixels (blue). **E** Absolute error of the Papenberg et al. algorithm for smoothing parameters $\alpha = 0.05$ (red) and $\alpha = 1$ (blue). **F** Regularized optic-flow field of the rubber-whale sequence with an angular error of ≈ 11 deg utilizing color information and confidence values. **G** Absolute error for the sequence presented in (A) as a function of noise level for our method (asterix), Lucas and Kanade with a window size of 1 pixel (square) and 5 pixels (diamond).

4 Conclusion

We proposed a novel algorithm for the computation of optic flow which is based on constructive interference of the global Fourier components of an image sequence. Since local information is reconstructed through superposition of global Fourier gratings, velocity estimates can be assigned to extended egdes and untextured parts of objects, as demonstrated for the translating-square sequence, *without* loosing spatiotemporal accuracy.

The performance of the algorithm has been evaluated for several synthetic and real image sequences. At low densities, low mean angular errors could be obtained for most sequences demonstrating that the algorithm is competitive with standard local algorithms for optic-flow computation (see Table 1). Smoothing may decrease the mean angular error while keeping a high density, but as a consequence, the local error at the motion boundaries increases drastically, causing a systematic error (see Fig. 5A-E). This effect is observed for many algorithms [2].

References

[1] E.H. Adelson and J.R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2: 284–299, 1985.

[2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV 2007*, 2007.

[3] J.L. Barron, D.J. Fleet, S. Beauchemin, and T. Burkitt. Performance of optical flow techniques. *Int. J. Comput. Vis.*, 12: 43-77, 1994.

[4] A. Briassouli, A. Narendra. Spatial and Fourier error minimization for motion estimation and segmentation. In *Proceedings of the 18th International Conference on Pattern Recognition*, ICPR 2006, August 20-24, Hong Kong, 2006.

[5] B.K. Dellen, J.W. Clark, and R. Wessel. The brain's view of the natural world in motion: Computing structure from function using directional Fourier transformations. *International Journal of Modern Physics B*, 21: 2493-2504, 2007.

[6] B. Galvin, B. McCane, K. Novins, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *Proc. British Machine Vision Conf.*, pages 195-204, 1998

[7] T. Gautama and M. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Trans. Neural. Net.*, 13: 1127-1136, 2003.

[8] N. Krüger and M. Felsberg. A continuous formulation of intrinsic dimension. In *Proceedings of the British Machine Vision Conference*, 2003.

[9] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. DARPA IU Workshop*, pages 121-130, 1981.

[10] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67:141-158, 2006.

[11] D. Vernon. Decoupling Fourier components of dynamic image sequences: a theory of signal separation, image segmentation, and optical flow estimation. In *Lect. Not. in Comp. Sci*, Springer Berlin/Heidelberg, Computer Vision – ECCV'98, pages 69, 1998.

[12] R. Wilson, G.H. Granlund. The uncertainty principle in signal processing. *IEEE Trans. Pat. An. Mach. Intel.*, 6:758–767, 1984.