Optical Flow Diffusion with Robustified Kernels.

Conference Paper · September 2005

Source: DBLP

CITATIONS
3

READS
79

2 authors:

Ashish Doshi
University of South Australia
17 PUBLICATIONS **196** CITATIONS
SEE PROFILE

Adrian G. Bors
The University of York
195 PUBLICATIONS **2,387** CITATIONS
SEE PROFILE

Optical Flow Diffusion with Robustified Kernels

Ashish Doshi and Adrian G. Bors

Dept. of Computer Science,
University of York, York YO10 5DD, United Kingdom
{adoshi, adrian.bors}@cs.york.ac.uk

Abstract. This paper provides a comparison study among a set of robust diffusion algorithms for processing optical flows. The proposed algorithms combine the smoothing ability of the heat kernel, modelled by the local Hessian, and the outlier rejection mechanisms of robust statistics algorithms. Smooth optical flow variation can be modelled very well using heat kernels. The diffusion kernel is considered Gaussian, where the covariance matrix implements the inverse of the local Hessian. Robust statistics operators improve the results provided by the heat kernel based diffusion, by rejecting outliers and by avoiding optical flow oversmoothing. Alpha-trimmed mean and median statistics are considered for robustifying diffusion kernels. The robust diffusion smoothing is applied onto multiple frames and is extended to 3D lattices.

1 Introduction

Analyzing motion patterns is essential for understanding visual surroundings [1]. When estimating motion, the common assumption is that the intensity of a moving pixel in the image plane is constant along its trajectory, in time [2]. This condition represents the main assumption for the optical flow constraint equation. However, in many cases, for example when representing the motion of fluids, the optical flow becomes very complex. Outlier vectors could affect the optical flow estimation in such situations.

Optical flow estimation algorithms can be classified as gradient-based and feature-based methods. A widely used local motion detection method is the block matching algorithm (BMA) [1]. BMA estimates the optical flow based on the correlation between a block of pixels in one frame and the corresponding block from within a search region in another frame and is used in the MPEG-2 motion compression standard. However, lack of contrast can lead to erroneous estimations. In order to overcome such problems, regularization terms have been used. Other approaches employ robust statistics algorithms [3,4].

This paper develops a methodology that combines the advantages of two different approaches: diffusion with heat kernels and robust statistics. This methodology is applied for smoothing vector fields. Perona and Malik introduced anisotropic diffusion for multiscale image segmentation [5]. Their proposed method uses partial differential equations (PDE) [6] to smooth grey-level images while preserving edges. Black et al. proposed *Tukey's biweight* function for obtaining

A. Gagalowicz and W. Philips (Eds.): CAIP 2005, LNCS 3691, pp. 222–230, 2005.

sharper boundaries [7]. Field regularization of orthonormal vector sets, using constraint-preserving anisotropic diffusion kernels was used by Tschumperlé and Deriche for denoising color images and for inpainting [8,9]. Their method introduces the use of tensor and Hessian matrices that are calculated from the local statistics. Important image structure such as edges and features are preserved while noise and smaller features are smoothed out in directions that are parallel with those of edges. Burgi [10] uses an optical flow constraint equation that includes diffusion terms and considers gradient directions. PDE's have been used in inpainting [9,11] and for various other applications [12].

The concept behind the approach presented in this paper is to enable the smoothing process with an outlier rejection mechanism. Diffused outliers can cause undesirable effects. The proposed algorithms aim to remove outliers without affecting structural data. When using local Hessian diffusion kernels, as in the case of the heat equation, the diffusion occurs along the direction of the edges, thus preserving the structure of objects. Section 2 introduces the application of diffusion kernels on vector fields and its extentions for 3D lattices. Section 3 introduces robust statistical diffusion kernels. Section 4 provides the experimental results of this study, while the conclusions are drawn in Section 5.

2 Hessian Diffusion Kernels

The heat equation of a geometric manifold can be described as [6]:

$$\frac{\partial I(x,t)}{\partial t} - \nabla^2 I(x,t) = 0 \tag{1}$$

where $I(x,t)$ represents the heat at location x and time t, starting with the initial conditions $I(x,0) = I(x)$ and ∇^2 denotes the Laplacian. The solution to the heat equation is [6]:

$$I(x,t) = \int_M K_t(x,y) I(y) dy \tag{2}$$

where $K_t(x,y)$ is the heat kernel (diffusion kernel) and $x, y \in M$. When $M \equiv I\!R$, the heat kernel becomes the Gaussian function :

$$I(x,t) = \frac{1}{\sqrt{4\pi d}} \int_{I\!R} \exp\left[-(\mathbf{x}-\mathbf{z}_c)^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\mathbf{z}_c)/4d\right] I(y) dy \tag{3}$$

where $\mathbf{\Sigma}$ represents the covariance matrix, $\mathbf{z}_c$ the kernel center and d is a normalization coefficient.

Differential techniques compute velocity from spatio-temporal derivatives of image intensities. After considering the Taylor series expansion, we obtain, [2]:

$$I(\mathbf{x}+\delta\mathbf{x}, t+\delta t) \approx I(\mathbf{x}, t+1) = I(\mathbf{x},t) + \nabla I \cdot \delta\mathbf{x} + \delta t g_t \tag{4}$$

where $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ and g_t represent first order partial spatial and temporal derivatives, respectively. Rearranging equation (4), we obtain :

$$\nabla I \cdot \mathbf{V} + g_t = 0 \tag{5}$$

where $\mathbf{V} = (V_x, V_y)$ denotes the motion vector and ∇I is the local intensity gradient. Equation (5) is known as the constrained optical flow equation. The optical flow is represented as a vector field on a 3D lattice. Each plane of the lattice corresponds to the motion between two consecutive frames [4]. Motion vectors can be calculated using the block matching algorithm. Block matching uses the correlation of a given image block from the frame $I(t)$ with blocks of the same size inside a search region in the next frame $I(t+1)$ [1]. Matching is performed by minimizing the displaced frame difference (DFD) for a particular block of pixels. Matching blocks in image areas that have constant grey-levels or similar texture patterns can lead to erroneous motion vectors [4]. The challenge is to achieve high robustness against strong assumption violations, commonly met in real sequences. In such cases, vector field smoothing is necessary. In order to obtain well defined moving objects and to maintain the optical flow constraints, second order differential methods are needed [2]. The local Hessian can be used as a detector of change in the direction of the optical flow.

The Hessian for the optical flow is represented as a matrix, $\mathbf{H}_{2D}$:

$$\mathbf{H}_{2D} = \begin{bmatrix} \psi_{xx} & \psi_{xy} \\ \psi_{yx} & \psi_{yy} \end{bmatrix} \tag{6}$$

whose entries ψ_{xx}, ψ_{xy}, ψ_{yx} and ψ_{yy} are second order partial spatial derivatives, $\psi_{xx} = \dfrac{\partial^2 V}{\partial x^2}$, $\psi_{yy} = \dfrac{\partial^2 V}{\partial y^2}$, $\psi_{xy} = \dfrac{\partial^2 V}{\partial x \partial y}$, $\psi_{yx} = \dfrac{\partial^2 V}{\partial y \partial x}$.

2.1 Smoothing Using 2D Hessian Kernel

The Hessian detects major changes in the direction of the optical flow. Optical flow associated with complex motion such as that of rotation, zooming or created by turbulent fluids can be accurately represented after being smoothed with a Hessian kernel. Hessians have been employed as kernels for diffusion smoothing in various applications [8,9]. The discretization of (3) is given by :

$$\hat{\mathbf{V}}_{kc}^{t+1} = \frac{\sum\limits_{\mathbf{x}_i \in \eta(\mathbf{z}_c)} \exp[-(\mathbf{x}_i - \mathbf{z}_c)^T \mathbf{H}_{2D,c}^{-1} (\mathbf{x}_i - \mathbf{z}_c)/4d] \cdot \mathbf{V}_{ki}^t}{\sum\limits_{\mathbf{x}_i \in \eta(\mathbf{z}_c)} \exp[-(\mathbf{x}_i - \mathbf{z}_c)^T \mathbf{H}_{2D,c}^{-1} (\mathbf{x}_i - \mathbf{z}_c)/4d]} \tag{7}$$

where $\mathbf{V}_{ki}^t$ is the vector at location i within a neighborhood $\eta(\mathbf{z}_c) = 3 \times 3$ around the central location $\mathbf{z}_c$, t denotes iteration number and k is the frame number.

2.2 Smoothing Using Multiple Frame 2D Hessian Kernel

The 2D Hessian kernel applies smoothing on pairs of 2 frames. This can be extended to include data from multiple frames, thus considering the temporal influence in smoothing. In this case the Hessian is calculated in each vector field, individually, as in (6), and the diffused vector takes into account all these Hessians assuming the temporal continuity. The resulting smoothed vector is :

$$\hat{\mathbf{V}}_{kc}^{t+1} = \frac{\sum_{j=k-K}^{j=k+K} \sum_{\mathbf{x}_i \in \eta(\mathbf{z}_{j,c})} \exp[-(\mathbf{x}_i - \mathbf{z}_{j,c})^T \mathbf{H}_{2D,jc}^{-1} (\mathbf{x}_i - \mathbf{z}_{j,c})/4d] \cdot \mathbf{V}_{ji}^t}{\sum_{j=k-K}^{j=k+K} \sum_{\mathbf{x}_i \in \eta(\mathbf{z}_{j,c})} \exp[-(\mathbf{x}_i - \mathbf{z}_{j,c})^T \mathbf{H}_{2D,jc}^{-1} (\mathbf{x}_i - \mathbf{z}_{j,c})/4d]} \tag{8}$$

where $j \neq k$ and $2K$ represents the number of frames considered.

2.3 Smoothing Using 3D Hessian Kernels

In the following the diffusion kernel is extended to 3D lattices :

$$\hat{\mathbf{V}}_{kc}^{t+1} = \frac{\sum_{\mathbf{x}_i \in \eta_{3D}(\mathbf{z}_c)} \exp[-(\mathbf{x}_i - \mathbf{z}_c)^T \mathbf{H}_{3D,c}^{-1} (\mathbf{x}_i - \mathbf{z}_c)/4d] \cdot \mathbf{V}_{ki}^t}{\sum_{\mathbf{x}_i \in \eta_{3D}(\mathbf{z}_c)} \exp[-(\mathbf{x}_i - \mathbf{z}_c)^T \mathbf{H}_{3D,c}^{-1} (\mathbf{x}_i - \mathbf{z}_c)/4d]} \tag{9}$$

where the neighbourhood is defined in 3D as $\eta_{3D}(\mathbf{z}_c) = 3 \times 3 \times 4$, and j is the central frame. In (9) the 2D Hessian kernel is extended to 3D to accommodate the spatio-temporal variation in the optical flow. By processing a larger amount of data, the optical flow transitions and moving object boundaries will be better preserved, whilst diffusing the vector field. The 3D Hessian matrix is given by :

$$\mathbf{H}_{3D} = \begin{bmatrix} \psi_{xx} & \psi_{xy} & \psi_{xt} \\ \psi_{yx} & \psi_{yy} & \psi_{yt} \\ \psi_{tx} & \psi_{ty} & \psi_{tt} \end{bmatrix} \tag{10}$$

where $\psi_{xx} = \dfrac{\partial^2 V}{\partial x^2}$, $\psi_{yy} = \dfrac{\partial^2 V}{\partial y^2}$, $\psi_{xt} = \dfrac{\partial^2 V}{\partial x \partial t}$, $\psi_{yt} = \dfrac{\partial^2 V}{\partial y \partial t}$, $\psi_{tx} = \dfrac{\partial^2 V}{\partial t \partial x}$, $\psi_{ty} = \dfrac{\partial^2 V}{\partial t \partial y}$ and $\psi_{tt} = \dfrac{\partial^2 V}{\partial t^2}$, where t denotes the frame index.

3 Robust Statistics Diffusion Kernels

3.1 Alpha-Trimmed Mean of Hessians Kernel

We integrate robust statistics into diffusion kernels. The alpha-trimmed mean algorithm, called also interquartile range averaging, ranks the given data and excludes from further computation a certain percentage of data at both extremes of the ranked array. The aim of this method is to remove outliers and to apply the diffusion algorithm only onto data that are statistically consistent. The updating equation is :

$$\hat{\mathbf{V}}_{kc}^{t+1} = \frac{\sum_{i=\alpha N}^{N-\alpha N} \exp[-(\mathbf{x}_i - \mathbf{z}_c)^T \mathbf{H}_c^{-1} (\mathbf{x}_i - \mathbf{z}_c)/4d] \cdot \mathbf{V}_{(i)}^t}{\sum_{i=\alpha N}^{N-\alpha N} \exp[-(\mathbf{x}_i - \mathbf{z}_c)^T \mathbf{H}_c^{-1} (\mathbf{x}_i - \mathbf{z}_c)/4d]} \tag{11}$$

where the motion vectors have been ranked according to their length, $\|\mathbf{V}_{(0)}\| < \|\mathbf{V}_{(1)}\| < \cdots < \|\mathbf{V}_{(N)}\|$, where $\|\mathbf{V}\|$ represents the length of the vector $\mathbf{V}$, N is the total number of vectors in the neighbourhood $\eta(\mathbf{z}_c)$, $\alpha \in (0,1)$ is the trimming percentage from a ranked array. The Hessian can be either $\mathbf{H}_{2D}$, multiple frame $\mathbf{H}_{2D}$ or $\mathbf{H}_{3D}$.

3.2 The Median of Directional Hessians Kernel

Another robust statistics approach consists of combining the use of median statistics with the diffusion kernel. Median algorithms have the ability to eliminate up to 50 % outliers and have been successfully used with radial basis function networks for moving object segmentation and motion estimation [4]. We first apply the Hessian-based diffusion algorithm as in (7) but calculated directionally, instead of centrally, with respect to the window location. We obtain directional smoothing for all the vectors from a certain neighbourhood. In this case we take into account extended neighbourhoods and we aim to reduce overlaps among local estimates. The total number of vectors considered in the smoothing process is extended from 3×3 vectors to 5×5 vectors. The median operator is applied onto the results produced by directional diffusions:

$$\mathbf{V}_{kc}^{t+1} = Median(\mathbf{V}_{kc}^{t+1}, \eta_{\mathrm{med}(\mathbf{z}_c)}) \tag{12}$$

where $\eta_{\mathrm{med}(\mathbf{z}_c)}$ is the window of the median operator centered at the location $\mathbf{z}_c$ that contains ranked diffused vectors. The influence of outliers will be diffused during the first operation. During the second processing operation any biased influence is eliminated. This operator can be applied with 2D Hessian (7), multiple frame 2D Hessian (8), or 3D Hessian (9) kernels.

4 Experimental Results

The proposed robust diffusion algorithms have been applied onto artificial vector fields as well as on optical flows extracted from image sequences. The vectorial field, entitled "Synthetic-1" is given by:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} D+S & -R \\ R & D-S \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} x-\mu \\ y-\mu \end{bmatrix} \tag{13}$$

where V_x and V_y are velocity components in the x and y direction, $c = \cos(\theta)$ and $s = \sin(\theta)$, respectively, where $\theta = 0$, $D = 0.8$ is the dilation coefficient, $S = 0.05$ is the shear coefficient, $R = 0.1$ is the rotation coefficient, and $\mu = 31$ is the center position of the resultant flow. The vector field, entitled "Synthetic-2," is created by differentiating the expression :

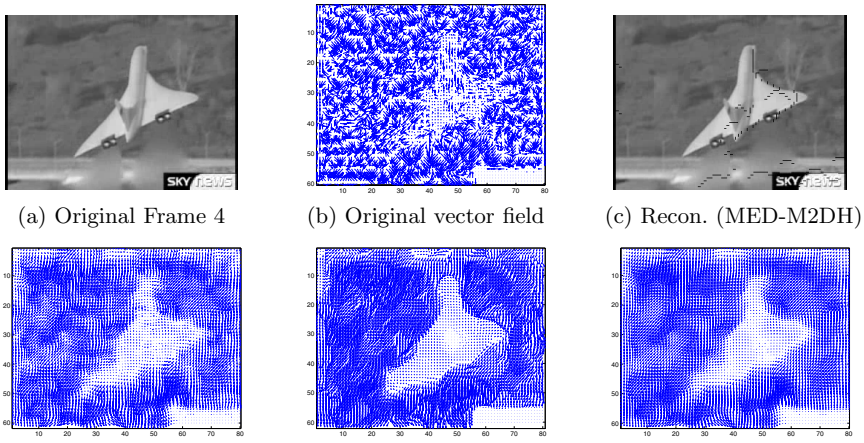
$$Z(x,y) = 3(1-x)^2 \exp^{-(x^2)-(y+1)^2} - 10(\frac{x}{5} - x^3 - y^5) \exp^{-x^2-y^2} - \frac{1}{3} \exp^{-(x+1)^2-y^2}$$

The velocity components are obtained as $\mathbf{V} = (\partial Z/\partial x, \partial Z/\partial y)$.

Table 1. MSE and MCE for synthetic vector fields after one iteration of diffusion

Data	Noise $\sigma 2$		Method									
			Perona-Malik		Black		2DH		ATM-2DH		MED-2DH	
			MSE	MCE	MSE	MCE	MSE	MCE	MSE	MCE	MSE	MCE
Synthetic-1	Gaussian	0.01	**0.007**	0.991	**0.007**	0.991	0.016	0.982	0.017	0.973	**0.007**	**0.995**
		0.10	0.063	0.933	0.063	0.933	0.097	0.931	0.154	0.860	**0.059**	**0.973**
		0.25	0.180	0.872	0.180	0.871	0.237	0.887	0.444	0.740	**0.126**	**0.952**
		0.30	0.229	0.855	0.229	0.854	0.276	0.873	0.528	0.721	**0.158**	**0.947**
		0.40	0.287	0.819	0.286	0.818	0.327	0.852	0.680	0.690	**0.186**	**0.933**
	Poisson	0.01	0.015	0.993	0.015	0.993	0.009	0.998	**0.001**	0.998	0.002	**0.999**
		0.05	0.309	0.963	0.309	0.963	0.182	0.978	**0.005**	**0.995**	0.094	0.987
		0.10	1.053	0.934	1.052	0.934	0.728	0.959	**0.036**	**0.985**	0.501	0.975
		0.25	7.594	0.797	7.594	0.797	7.091	0.830	**1.627**	**0.867**	6.454	0.851
		0.40	20.874	0.649	20.873	0.649	20.767	0.668	**10.629**	**0.712**	19.470	0.692
Synthetic-2	Gaussian	0.01	**0.011**	0.722	**0.011**	0.722	0.019	0.682	0.023	0.626	0.012	**0.776**
		0.10	0.061	0.532	0.061	0.532	0.088	0.531	0.239	0.428	**0.043**	**0.612**
		0.25	0.156	0.435	0.156	0.435	0.195	0.473	0.473	0.350	**0.108**	**0.554**
		0.30	0.214	0.469	0.213	0.469	0.253	0.493	0.482	0.362	**0.146**	**0.572**
		0.40	0.425	0.415	0.425	0.414	0.522	0.443	0.658	0.320	**0.342**	**0.546**
	Poisson	0.01	0.021	0.961	0.021	0.961	0.027	0.956	**0.007**	0.974	0.008	**0.985**
		0.05	0.291	0.803	0.291	0.805	0.338	0.800	**0.013**	**0.969**	0.113	0.889
		0.10	1.364	0.631	1.364	0.633	1.482	0.633	**0.070**	**0.923**	0.832	0.703
		0.25	8.008	0.374	8.008	0.376	8.106	0.372	**2.427**	**0.623**	6.883	0.359
		0.40	19.592	0.239	19.591	0.240	20.073	0.237	**10.871**	**0.359**	17.994	0.219

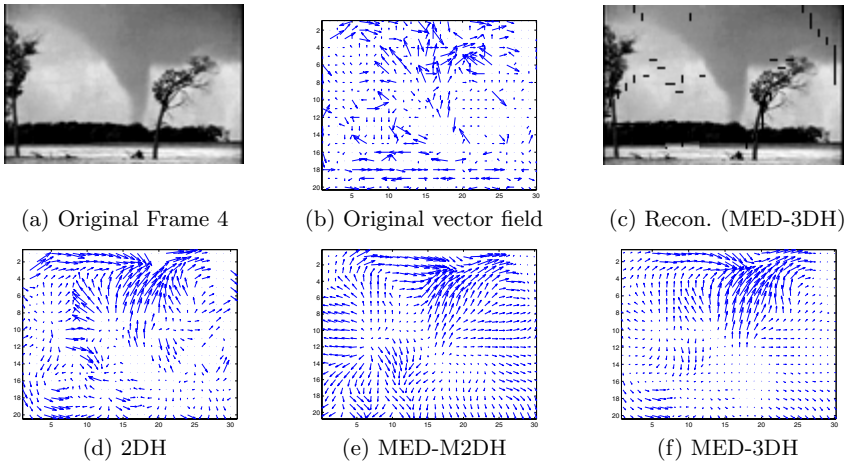
We consider Gaussian and Poisson noise distributions, each with five different variances, corrupting the given vector fields. The algorithms described in this paper and two well know diffusion algorithms, respectively Perona-Malik (PM) [5], and Black [7], that have been adapted for the use on vectorial data, are applied for smoothing the noisy vector fields with the aim of trying to reconstruct the original vectorial fields. The numerical results obtained after smoothing the synthetic vectorial fields corrupted by noise after one iteration by the given algorithms are provided in Table 1. The algorithms are denoted according to the type of kernel and the algorithm that has been used for smoothing as: 2DH - 2D Hessian, ATM-2DH - alpha trimmed mean using 2D Hessian, M2DH - multiple 2D Hessian, ATM-M2DH - alpha trimmed mean of multiple 2D Hessian, 3DH - 3D Hessian, ATM-3DH - alpha trimmed mean of 3D Hessian, MED-2DH - median of 2D Hessian, MED-M2DH - median of multiple 2D Hessian, MED-3DH - median of 3D Hessian. For the alpha-trimmed mean smoothing algorithm in the case of the H_{2D} kernel we consider $N = 9$, while $\alpha N = 3$ and so 6 vectors are eliminated from the diffusion process. The results are assessed numerically in terms of mean square error (MSE) and mean cosine error (MCE). The second measure calculates the average cosines of the angle between the ground truth vector and its smoothed version. In Table 1 the best results are highlighted. MED-2DH proved to be better in the case when removing the Gaussian noise, while ATM-2DH is better in the case of Poisson noise.



(a) Original Frame 4 (b) Original vector field (c) Recon. (MED-M2DH)

(d) M2DH (e) ATM-M2DH (f) MED-M2DH

Fig. 1. Results obtained for "Concorde" sequence



(a) Original Frame 4 (b) Original vector field (c) Recon. (MED-3DH)

(d) 2DH (e) MED-M2DH (f) MED-3DH

Fig. 2. Results obtained for "Tornado" sequence

The second set of experiments provides a comparison when diffusion algorithms are applied on optical flows estimated from image sequences. The block matching algorithm (BMA) has been used to estimate motion vector fields in a set of image sequences. The image sequences considered are listed in Table 2. Fig. 1 shows the results obtained when applying diffusion algorithms on "Concorde take-off" sequence, while Fig. 2 shows the results when processing the "Tornado" sequence. Fig. 1a displays frame 4 of the "Concorde" sequence, Fig. 1b shows the vector field extracted between frames 4 and 6, using BMA. This sequence was chosen for its complex motion characteristics such as rotational move-

Table 2. PSNR between the predicted frame based on smoothed optical flow after 1 iteration and the actual frame

Method	PSNR(dB)					
	Taxi	Concorde	Fighter	Clouds	Tornado	Traffic
PM	18.15	17.40	16.65	16.93	21.14	12.97
Black	18.05	17.39	16.68	16.99	21.21	13.00
2DH	19.10	17.83	17.32	17.57	21.72	13.90
M2DH	19.43	19.03	18.23	16.91	21.75	13.67
ATM-2DH	20.40	16.81	16.62	16.85	22.32	14.05
MED-2DH	20.99	19.04	19.13	20.29	22.95	15.84
ATM-M2DH	20.12	18.59	16.20	16.86	23.33	13.56
MED-M2DH	20.88	**20.40**	**20.07**	**20.31**	**23.71**	**16.10**
3DH	18.96	17.77	17.29	17.76	21.64	13.86
ATM-3DH	19.84	18.31	16.05	16.54	23.40	13.30
MED-3DH	**21.07**	19.00	19.10	20.28	23.09	15.83

ment, turbulent air from jet thrusters, blocky artifacts from compression and camera movement combined with a rigid moving object. In order to assess the efficiency of the smoothed optical flows we calculate the PSNR (peak signal-to-noise ratio) between the next frame and the corresponding frame reconstructed using the smoothed optical flow. Fig. 1c shows the reconstructed frame 6, obtained by translating individually each block of the frame with its corresponding smoothed vector by MED-M2DH. Figs. 1d-1f show the smoothed vector fields after 5 iterations when using M2DH, ATM-M2DH and MED-M2DH, respectively. The corresponding results are displayed in Fig. 2 for the "Tornado" sequence. The movement and the area affected by the twister can be easily identified after employing the robust diffusion algorithms and their multi frame extensions, as it can be seen in Figs. 2d-2f. In the case of the 3D Hessian kernel the total number of vectors considered in smoothing is $N = 36$.

Table 2 provides the PSNR calculated between the actual frame and the predicted frame for six different image sequences. From this table as well as from Figs. 1 and 2 it can be observed that robust kernels such as MED-M2DH and MED-3DH, have better performance than Perona-Malik, Black or 2DH algorithms. Methods that use data from the 3D lattice such as those based on ATM-3DH and MED-3DH kernels smooth better than 2DH kernels.

5 Conclusions

A set of robust diffusion algorithms is proposed for optical flow smoothing. The diffusion kernel ensures that smoothing occurs along the optical flow structure. The extention of 2D Hessian to the 3D Hessian kernel considers the temporal information from multiple frames. Robust statistics algorithms such as alpha trimmed-mean and marginal median are employed on diffusion kernels for removing the outliers and for enhancing vector smoothing. The improvements

provided by the robust algorithms are particularly evident when dealing with complex optical flows such as those that describe the motion of fluids. Optical flow smoothing algorithms represent a core processing module for motion estimation, segmentation, tracking of moving objects as well as in video compression.

References

1. A. M. Tekalp, *Digital Video Processing.* Prentice Hall, 1995.
2. J. L. Barron, D. J. Fleet, S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43-77, 1994.
3. M. Black, P. Anandan, "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, Vol. 63, No. 1, pp. 75-104, 1996.
4. A. G. Bors, I. Pitas, "Optical Flow Estimation and Moving Object Segmentation Based on Median Radial Basis Function Network," *IEEE Trans. on Image Processing,* vol. 7, no. 5, pp. 693-702, 1998.
5. P. Perona, J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 12, no. 7, pp. 629-639, 1990.
6. S.-T. Yau, ed., *Surveys in Differential Geometry: Differential Geometry inspired by String Theory.* American Mathematical Society, 1999.
7. M. J. Black, G. Sapiro, D. H. Marimont, David Heeger, "Robust Anisotropic Diffusion," *IEEE Trans. on Image Processing*, vol. 7, no. 3, pp. 421-432, 1998.
8. D. Tschumperlé, R. Deriche, "Orthonormal vector sets regularization with PDE's and Applications," *Inter. Jour. on Computer Vision*, vol. 50, no. 3, pp. 237-252, 2002.
9. D. Tschumperlé, R Deriche, "Vector-Valued Image Regularization with PDE's: A Common Framework for Different Applications," *Proc. IEEE Computer Vision and Pattern Recognition*, Madison, USA, 2003, vol. I, pp. 651-656.
10. P.-Y. Burgi, "Motion estimation based on the direction of intensity gradient," *Image and Vision Computing*, vol. 22, no. 8, pp. 637-653, 2004.
11. M. Bertalmio, L. Vese, G. Sapiro, S. Osher, "Simultaneous Structure and Texture Image Inpainting," *IEEE Trans. on Image Proces.*, vol. 12, no. 8, pp. 882-889, 2003.
12. M. Irani, "Multi-Frame Optical Flow Estimation Using Subspace Constraints," *Proc. IEEE Int. Conf. on Computer Vision*, Corfu, Greece, 1999, vol. I, pp. 626–633.