The minors of the structure tensor

Erhardt Barth

Institute for Signal Processing
University of Lübeck, Ratzeburger Allee 160, 23538 Lübeck
barth@isip.mu-luebeck.de http://www.isip.mu-luebeck.de

Abstract. A novel method for motion estimation from first order derivatives is presented. First estimates are obtained by evaluating the minors of the so-called structure tensor that contains blurred products of first order derivatives. The minors yield four different estimates that are equal in case of translation but differ for other spatio-temporal patterns. The mean yields a robust motion estimate and the difference is used as an indicator of discontinuous motions, occlusions, and noise. This procedure leads to a flow field with small errors and low density. An additional change detection is performed to obtain a mask that is filled with the previously computed (correct but sparse) motion vectors. The superior performance of the algorithm is demonstrated on synthetic and real sequences by comparison with other methods.

Keywords: image-sequence analysis, motion, flow field, curvature, dynamic features, change detection

1 Introduction

We consider image sequences defined by intensity $f(x,y,t)$ and construct the following matrix from the first-order derivatives of f:

$$\mathbf{D}(x,y,t) = (f_x,f_y,f_t)^T(f_x,f_y,f_t) = \begin{pmatrix} {f_x}^2 & f_xf_y & f_xf_t \\ f_xf_y & {f_y}^2 & f_yf_t \\ f_xf_t & f_yf_t & {f_t}^2 \end{pmatrix}. \tag{1}$$

Since this matrix obviously does not contain more information than the gradient (f_x,f_y,f_t) itself, a different matrix, obtained from $\mathbf{D}$ by convolution with a smoothing kernel $h(x,y)$ (or $h(x,y,t)$), i.e.

$$\mathbf{J}(x,y,t) = h(x,y) * \mathbf{D}(x,y,t), \tag{2}$$

can be used to characterize the structure of $f(x,y,t)$. Accordingly, $\mathbf{J}$ has been called "the structure tensor". In the context of motion estimation, it has been shown how the eigenvectors of $\mathbf{J}$ can be used to estimate the optical flow. The benefit of using $\mathbf{J}$ is that useful measures, which indicate confidence in the motion estimate, can be defined based on the eigenvalues of $\mathbf{J}$ - see [5,6] for a review.

Related methods have been proposed that are based on the geometry of the hypersurface $\boldsymbol{S} = (x,y,t,f(x,y,t))$, e.g. for the purpose of motion detection [8].

Further, it has been shown that the Gaussian curvature of this hypersurface can be used to detect motion discontinuities [14] and that the Riemann curvature tensor $\mathbf{R}$ can be used to estimate motion parameters [2]. Four different expressions, based on second order partial derivatives of f, for the motion vector have been derived from $\mathbf{R}$ and differences among these vectors have been used as confidence measures. The expressions derived for the components of $\mathbf{R}$ also hold for the minors of the 3D Hessian of f. In this paper we will derive similar expressions for the minors of the matrix $\mathbf{J}$.

Covariance-based methods have been proposed as alternatives to differential methods [4]. $\mathbf{J}$ approximates the covariance matrix $\mathbf{C}_g$ of the gradient (f_x, f_y, f_t) and it is argued that, in 2D, $\mathbf{C}_g$ should be computed as the Hessian of the autocovariance of f [10]. From this perspective, methods based on $\mathbf{R}$ versus $\mathbf{J}$ differ in that they are based on the Hessian of f, and of the autocovariance of f, respectively. Similarly, one could use the Hessian of local-energy band-pass filter outputs or other representations of f. In [2] it had been suggested to estimate motion from the 3D Hessian of a ratio-of-Gaussians filter output.

Various methods for motion estimation are known and comprehensive reviews can be found, e.g. in [7, 6, 12]. The comparative analysis presented here, however, is limited to the methods by Lukas and Kanade (L&K) [9], Uras et al. (Uras) [13], the structure-tensor method (ST), the new method based on the minors of the structure tensor (MST), and an extended version thereof (filled MST).

2 Theory

We now consider the minors of $\mathbf{J}$, i.e. the matrix

$$\mathbf{M} = \text{Minors}(\mathbf{J}). \tag{3}$$

The elements $M_{ij}, (i, j = 1, 2, 3)$ of $\mathbf{M}$ are the determinants of the matrices obtained from $\mathbf{J}$ by eliminating the row $4-i$ and the column $4-j$, e.g., $M_{11} = (h*{f_x}^2)(h*{f_y}^2) - (h*(f_x f_y))^2$. The results presented in this section were obtained with the following analytical expression for $\mathbf{J}$:

$$\mathbf{J}(x, y, t) = \sum_{i,j=0}^{N-1} w_{ij} \mathbf{D}(x + i, y + j, t), \tag{4}$$

i.e., sums of functions $\mathbf{D}(x, y, t)$ shifted with integer (for simplicity) amounts in x and y have been simplified. The results do not depend on the weights w_{ij} (they cancel out) and not on N (note, however, that for $N = 1$, i.e. $\mathbf{J} = \mathbf{D}$ all the minors are zero).

2.1 Translation with constant velocity

If the image sequence $f(x, y, t)$ results from any spatial pattern moving with constant velocity $\boldsymbol{v} = (v_x, v_y)$, f is supposed to satisfy the constraint

$$f(x, y, t) = f(x - v_x t, y - v_y t). \tag{5}$$

Under this constraint we obtain the following relations for the minors of $\mathbf{J}$:

$$\begin{aligned}(M_{31}, -M_{21})/M_{11} &= \boldsymbol{v}_1 \\ (M_{23}, -M_{22})/M_{12} &= \boldsymbol{v}_2 \\ (M_{33}, -M_{23})/M_{13} &= \boldsymbol{v}_3 \\ (M_{33}, -M_{22})/M_{11} &= (v_{4x}^2, v_{4y}^2).\end{aligned} \tag{6}$$

Indices $i = 1, ..., 4$ simply denote the fact that we obtain different expressions for $\boldsymbol{v}$. All representations $\boldsymbol{v}_i$ were obtained by assuming the constraint in Eq. 5 and by performing the simplification of all possible ratios of $\mathbf{M}$ components. [1]

Obviously, $\boldsymbol{v}_4$ can be computed as $(\text{sign}(\boldsymbol{v}_{1x})\sqrt{M_{33}}, \text{sign}(\boldsymbol{v}_{1y})\sqrt{-M_{22}})/\sqrt{M_{11}}$ to account for the sign of motion. To summarize, we found four different combinations of minors that are equal and equal to the motion vector in case that Eq. 5 holds ($\boldsymbol{v} = \boldsymbol{v}_1 = \boldsymbol{v}_2 = \boldsymbol{v}_3 = \boldsymbol{v}_4$).

2.2 Translation with time-dependent velocity

We now consider the more general case where the image shift contains higher-order terms, i.e., the motion can be accelerated:

$$f(x, y, t) = f(x - d_1(t), y - d_2(t)). \tag{7}$$

For the minors of $\mathbf{J}$ under the constraint (7) we obtain the same results as in Eq. 6 with $\boldsymbol{v}_i = (d_1'(t), d_2'(t))$. This is an interesting result since it cannot be obtained for the minors of the Hessian.

2.3 On- and offset of spatial patterns

Here we consider the cases: $f(x, y, t) \rightarrow f(x, y)\gamma(t)$ and $f(x, y, t) \rightarrow f(x, y)\delta(t)$ with the step function $\gamma(t)$ and the Dirac-delta distribution $\delta(t)$. In these cases, we do not obtain any meaningful simplifications. However, the results differ for the four expressions $\boldsymbol{v}_i$ obtained for the motion vector $\boldsymbol{v}$ in the previous sections. Therefore the difference can be used as an indicator of pattern on- and offset that typically occurs with occlusions - see Section 3.3.

3 Algorithms

3.1 Traditional differential methods

The algorithms by Uras et al. [13] and Lukas and Kanade [9] were implemented by using the code provided by the authors of [7] and were applied with the original parameter settings (temporal filtering with a sigma of 1.5 pixels in both cases and spatial filtering with a sigma of 3 pixels and 1 pixel, respectively).

[1] These and the following simplifications have been performed with the aid of the software *Mathematica*; a formal proof will be given in a forthcoming paper.

3.2 Structure tensor (ST algorithm)

Here we used our own implementation to enable a direct comparison with the MST method (see below), i.e., the computation of **J** was a common block in both the ST and the MST algorithm. The sequence was first low-pass filtered with a spatio-temporal Gaussian filter (cutoff frequencies 0.33 in spatial frequency and 0.75 in temporal frequency - both values given in fractions of the maximal frequency). Subsequently, the components of **J** were computed by convolution with discrete kernels (-1,0,1), multiplication, and subsequent convolution with a Gaussian smoothing kernel $h(x, y)$ (Eq. 2) with a sigma of 2 pixels. The computation of **J** can be further optimized [6] but we were interested in a simple implementation (that offers the possibility of a straightforward and optimal multi-scale extension) and a performance analysis relative to the MST algorithm. The following operations were performed only in regions were the minor $M_{11} > T_{M_{11}}$ with $T_{M_{11}}$ set to one percent of the maximum of M_{11} in each frame (the MST algorithm uses this confidence measure to avoid division by zero). We then computed the eigenvalues λ_i of **J** and sorted them $\lambda_1 > \lambda_2 > \lambda_3$. In cases where λ_1 was larger than a threshold T_λ,[2] and the confidence measure $c_c = ((\lambda_1 - \lambda_3)/(\lambda_1 + \lambda_3))^2 - ((\lambda_1 - \lambda_2)/(\lambda_1 + \lambda_2))^2$ greater than T_c, we computed the eigenvector $\boldsymbol{r}_3 = (r_{3,1}, r_{3,2}, r_{3,3})$ corresponding to λ_3. In these cases, the motion vector was set to $\boldsymbol{v}_{ST} = h_v(x, y) * ((r_{3,1}, r_{3,2})/r_{3,3})$. The thresholds were chosen $T_\lambda = 20$ and $T_c = 0.4$ such as to optimize the tradeoff between density and noise behavior, and to obtain a comparable density for the two methods ST and MST. The Gaussian smoothing kernel h_v had a sigma of 2 pixels.

3.3 Minors of the structure tensor (MST algorithm)

Starting with **J** as defined above, the vectors v_i were computed according to Eq. 6 at those locations, where the respective denominators were larger than one percent of their maximum value at that frame. If ${v_{ix}}^2 + {v_{iy}}^2 > {T_m}^2$ and the angular deviation among the four vectors $\boldsymbol{v}_i$ less than T_θ degrees, the spatially blurred mean of the four was taken as the final result, i.e., $\boldsymbol{v}_{MST} = h_v(x, y) * (v_1 + v_2 + v_3 + v_4)/4$. Otherwise $\boldsymbol{v}_{MST}$ was set to zero. The value of T_m was 5 percent of the maximum length of the motion-vector $\boldsymbol{v}_1$ (at each frame) in case of the synthetic sequence and 1 percent for the traffic scene (this criterion could not be used with the ST algorithm because it eliminated correct vectors due to large maxima). The values of T_θ were 4 and 15 degree, and the Gaussian smoothing kernel h_v had a sigma of 2 (as for ST) and 4 pixels, respectively.

3.4 Change-detection and filling (filled-MST algorithm)

We used a change-detection algorithm based on a Bayesian decision criterion [1]. The absolute difference of two consecutive frames is summed in a 5×5 neighborhood and normalized by the noise standard deviation in stationary regions. Pixels where this normalized sum is above a threshold are set to one and indicate the regions with temporal change. The other pixels are set to zero. In addition,

[2] In the printed version, this erroneously read "[...] where λ_3 was larger [...]".

the threshold is changed adaptively such as to increase the probability for label "1" by the number of spatial neighbors that have already been labeled "1".

The above mask is then filled with the initially obtained motion vectors as follows. If a pixel has (i) a mask value of one, (ii) a zero motion vector, and (iii) neighbors with non-zero motion vectors, it is assigned the mean vector of the (non-zero) neighbors. All other pixels remain unchanged. This procedure is repeated until the mask is filled (the number of iterations was 30 for the square and 22 for the taxi). Since all the displayed flow fields were sub sampled for graphical reasons (by factors 6 in Fig. 2 and 4 in Fig. 3) after low-pass filtering to avoid aliasing, they contain an additional smoothing (that introduces some errors). The error analysis, however, was computed without smoothing.

4 Results

4.1 Synthetic sequence with noise

We used an image sequence of size 256×256 pixels that was 64 frames long. In this sequence, a gray square (intensity 128) of size 64×64 pixels appears against a darker background (intensity 64) at frame 23, moves 2 pixels right and one pixel up in each frame and then disappears at frame 44. To this sequence, uniformly distributed noise with a low variance of 7 was added. In addition, to randomly chosen blocks of size $4 \times 4 \times 4$ pixels in space and time the value of 54 was added or subtracted. This procedure is supposed to simulate the appearance and disappearance of light and dark blobs. We have chosen this input to test the algorithms with local discontinuities (occlusions) and aperture problems.

Comparative results obtained for this input are shown in Fig. 1. The top left plot shows the angular errors (standard deviation of estimated motion direction compared to the true motion evaluated for the pixels that belong to the moving square) as a function of frame number. Note that the errors are lowest for the MST algorithms (continuous lines). The top right plot illustrates the relative number of pixels in the background for which motion vectors were estimated (incorrectly since the background does not move). Note that in this plot the continuous curve (MST) hardly deviates from zero and that the ST algorithm produces similarly good results. The bottom left graph plots (on a log scale) the density of the flow field (the relative number of pixels for which a motion vector was estimated). A similar plot that shows the density obtained by counting only the vectors with an angular error of less than 8 degree is shown bottom right. Note that the ST and MST methods obtain the higher acuity at the expense of a lower density. Also note, however, that for all methods other than filled MST the density is low, reflecting the fact that reliable estimates can be obtained only at the corners of the square. For the filled MST method the increased density is obtained at the expense of a somewhat higher number of erroneous motion vectors (due to filling a noisy mask). Sample flow fields of the synthetic sequence are shown in Fig. 2. The left panel displays results for frame number 23, when the square appears. Note that for the MST-based algorithms all vectors are zero

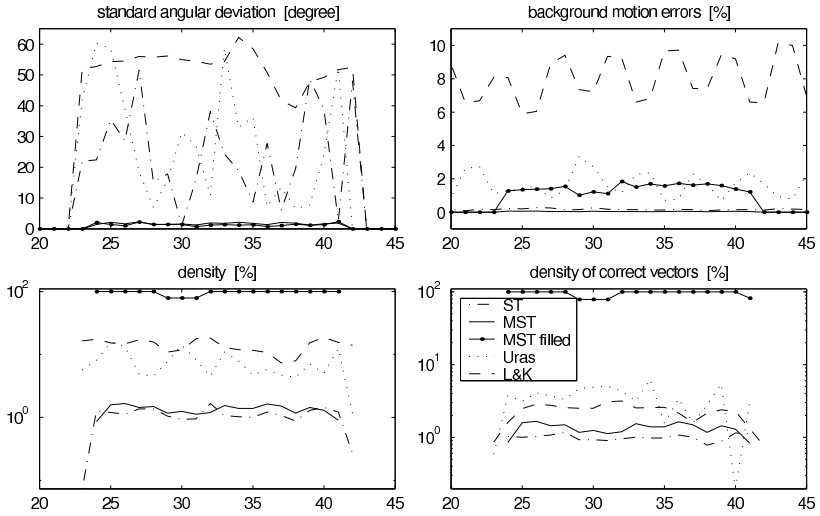

Fig. 1. Results for the synthetic sequence - see text.

(as the motion is). The other algorithms (especially Uras and L&K) compute false motion vectors induced by the popping square and the noise.

The same type of results are shown in in the right panel of Fig. 2 for frame number 34 when the square translates. Note that in this case the ST and MST algorithms yield a certain number of correct motion vectors at the corners of the square and minimize the effect of the noise. Note also that the filled-MST algorithm produces a much higher density (at the expense of some additional errors). Of course, the filling algorithm could be applied to other methods also, but it is crucial that all incorrect motion vectors are eliminated prior to filling.

4.2 Traffic scene

Results obtained with the filled-MST algorithm for a real video sequence are shown in Fig. 3. Note the high density obtained by change detection and filling. Also note, that the pedestrian is detected correctly in the top left corner. This example illustrates that the filled MST algorithm performs well on real sequences; a more comprehensive analysis is desirable but beyond the scope and size of this paper.

Since different implementations have been used, a direct comparison of computation time is only meaningful for the ST and MST algorithms. In our implementations (MATLAB) the ST algorithm took $0.52 msec$ ($244 flops$) per pixel and the MST algorithm $0.04 msec$ ($43 flops$). The filled MST algorithm is much slower but the speed of change detection can be improved. Also, filling will be replaced by more efficient reconstruction algorithms: since it has been shown [4] and proved [11] that images can be reconstructed from curved regions only, the aperture and density problems do (theoretically) not exist.

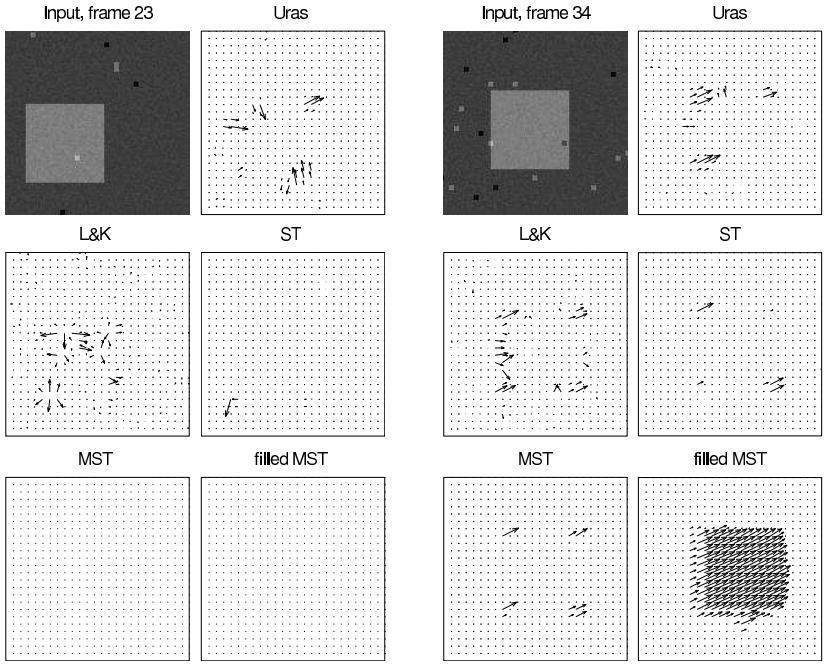

Fig. 2. Sample results for frame 23 when the square appears (left panel) and frame 34 when the square translates (right panel).

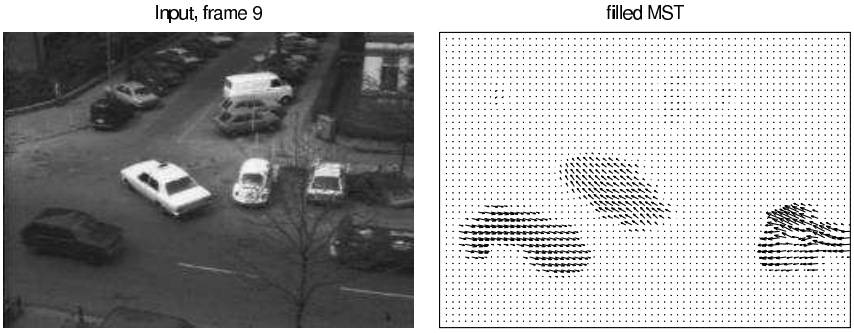

Fig. 3. Hamburg-taxi scene and results of filled-MST algorithm.

5 Conclusions

We have presented a novel method for motion estimation based on the minors of the structure tensor $\mathbf{J}$. The results are better than those obtained by computing the eigenvalues and eigenvectors of $\mathbf{J}$ (and better than those based on the minors of the Hessian of f). Traditional first- and second-order differential methods are also outperformed.

The key results and features are that (i) the minors of $\mathbf{J}$ yield four different expressions $\boldsymbol{v}_i$ for the motion vector (ii) these expressions are equal for translations and accelerated motions but differ for occlusions and noise (iii) the flow field that results from accepting only similar $\boldsymbol{v}_i$'s is correct but sparse (iv) the density can be increased substantially by combining motion estimation with change detection.

6 Acknowledgment

We thank Amir Madany Mamlouk and Thomas D. Otto for their high engagement in the simulations, and D. Toth for providing the change-detection algorithm. We also thank C. Mota and the reviewers for comments on the manuscript.

References

1. T. Aach and A. Kaup. Bayesian algorithms for adaptive change detection in image sequences using Markov random fields. *Signal Processing: Image Communication*, 7(2):147–160, 1995.
2. J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994.
3. E. Barth. Bewegung als intrinsische Geometrie von Bildfolgen. In W. Förster, J. M. Buhmann, A. Faber, and P. Faber, editors, *Mustererkennung 99*, pages 301–308, Bonn, 1999. Springer, Berlin.
4. E. Barth, T. Caelli, and C. Zetzsche. Image encoding, labelling and reconstruction from differential geometry. *CVGIP:GRAPHICAL MODELS AND IMAGE PROCESSING*, 55(6):428–446, 1993.
5. J. Berkmann and T. M. Caelli. Computation of surface geometry and segmentation using covariance techniques. *IEEE: Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1994.
6. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer, 1995.
7. B. Jähne, H. Haußecker, and P. Geißler, editors. *Handbook of Computer Vision and Applications*, volume 2, chapter 13. Academic Press, Boston, 1999.
8. S.-P. Liou and R. C. Jain. Motion detection in spatio-temporal space. *Computer Vision, Graphics, and Image Processing*, 45:227–50, 1989.
9. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, pages 121–130, 1981.
10. R. Mester. Orientation estimation: conventional techniques and a new non-differential approach. In *Proceedings EUSIPCO 2000*.
11. C. Mota and J. Gomes. Curvature operators in geometric image processing. In *Brazilian Symposium on Computer Graphics and Image Processing*. IEEE Press, 1999.
12. H. H. Nagel. On the estimation of optical flow: relations between different approaches and some new results. *Artificial Intelligence*, 33:299–324, 1987.
13. S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion computation. *Biological Cybernetics*, 60(5):79–97, 1988.
14. C. Zetzsche and E. Barth. Direct detection of flow discontinuities by 3D curvature operators. *Pattern Recognition Letters*, 12:771–9, 1991.