Multiple Motion Estimation using Channel Matrices

Per-Erik Forssén[1] and Hagen Spies[2]

[1] Computer Vision Laboratory, Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden
[2] R&D ContextVision AB, Storgatan 24
SE-582 23 Linköping, Sweden

Abstract. The motion field from image sequences of a dynamic 3D scene is in general piecewise continuous. Since two neighbouring regions may have completely different motions, motion estimation at the discontinuities is problematic. In particular spatial averaging of motion vectors is inappropriate at such positions. We avoid this problem by channel encoding brightness change constraint equations (BCCE) for each spatial position into a channel matrix. By spatial averaging of this channel representation and subsequently decoding we are able to estimate all significantly different motions occurring at the discontinuity, as well as their covariances. This paper extends and improves this multiple motion estimation scheme by locally selecting the appropriate scale for the spatial averaging.

1 Introduction

The motion field from image sequences of a dynamic 3D scene is in general piecewise continuous. Since two neighbouring regions may have completely different motions, motion estimation at the discontinuities is problematic. In particular this means that linear estimation of motion parameters in regions containing a boundary is inappropriate. Furthermore, effects such as shadows and transparency can result in there actually being several valid motions at a single image location [1].

The problem of smoothing across edges can be solved to some extent by doing motion estimation in a small neighbourhood, and then applying edge preserving filtering, or robust estimation techniques to the resultant motion field. This will reduce noise, and eliminate outliers in the initial measurements, but the required size of the local region will vary considerably due to the *aperture problem* (the motion in an intrinsic-1D neighbourhood is ambiguous, see e.g. [1, 2]). Furthermore we will still run into problems if the local region contains several valid motions, as in the case of transparency and thin elongated objects such as tree branches.

In order to make the initial linear estimation region even smaller, one could instead replace the initial motion estimation step with a *motion constraint* estimation step. A popular motion constraint is the brightness change constraint equation (BCCE), which relates spatial and temporal derivatives (f_x, f_y, f_t) of the signal f, with the local image plane motion $\begin{bmatrix} u \; v \end{bmatrix}^T$

$$uf_x(x, y) + vf_y(x, y) + f_t(x, y) = 0\,. \tag{1}$$

The BCC-equation is based on the assumption of constant intensity, and can be derived from a first order Taylor expansion of a signal undergoing an infinitesimal translation. Since (1) is valid also in regions where the aperture problem is present, it can be correctly estimated using much smaller spatial windows. By clustering BCCEs in the u–v plane within a local region we can then estimate several local image plane motions. Examples of this approach are e.g. [1, 3], where the EM algorithm has been used to do the clustering.

The BCCE is actually an incorrect motion model in a number of situations: 1) if the illumination changes, 2) at the occlusion boundary when the background is non-constant, 3) if two motions are present, as is the case at e.g. moving shadow boundaries and reflections.

We have previously [4] developed a clustering technique which can automatically reject most of the incorrect motion constraints, and estimate several solutions to a system of constraints in a local neighbourhood, by encoding motion constraint estimates in *channel matrices*, performing spatial averaging of the matrix elements, and then decoding. Averaging the channel matrices adds the assumption that the motion is locally constant.

There have been several other attempts to determine multiple motions, e.g. [5–8]. For a discussion of the Fourier properties of multiple motions see [9]. Often some type of filter bank is used where the filter outputs are either combined into a multiple motion likelihood function or used as separate constraints in an over-determined system of equations. In the presented approach we use simple derivative filters to yield the constraint equation that is input into our estimation scheme.

1.1 Organisation of paper

This paper is organised as follows: In section 2, we describe the channel representation of motion constraints, and conversion to and from it. In section 3, we describe how the channel representation can be used to estimate optical flow, and demonstrate the behaviour of the algorithm using the well known “Hamburg taxi”, and a synthetic sequence. In section 4 we introduce an algorithm which locally adapts the size of the region in which motion is estimated, and in section 5 we compare this method to least-squares optical flow using the “flower garden” sequence.

2 Channel representation

Channel representation [10–12] is a technique to represent single or multiple statements with associated confidences in a uniform manner. Channel representation has applications in learning, clustering and edge-preserving filtering [10].

In the channel representation, a measurement u, and its confidence r are represented as a vector $\boldsymbol{\Phi}$ of K channel values Φ_k. The channel values are computed by passing the measurement u through a set of shifted *kernel functions* $g(u-k)$, and weighting the result with the confidence[3] r, i.e. $\Phi_k(u, r) = rg(u-k)$. Averaging in the channel representation followed by a *local decoding* is a way to estimate the modes of the PDF $p(u)$ [10].

Common kernel choices are $\cos^2$, B-spline, and Gaussian kernels. In this paper we will use Gaussians, since decoding of a Gaussian channel vector allows recovery of both mode location and standard deviation.

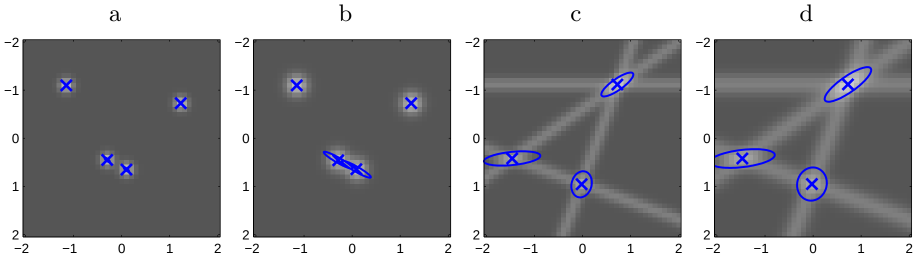

Fig. 1. Example of channel histograms with sampling distance $\Delta u = 0.1$. a) and b) Encoding of points with $\sigma = \Delta u$ and $\sigma = 1.5\Delta u$, c) and d) Encoding of lines with $\sigma = \Delta u$ and $\sigma = 1.5\Delta u$. Decoded mode locations are visualised as crosses, and the covariances as ellipses.

2.1 Encoding of Points

When encoding a point $\begin{bmatrix} u\ v \end{bmatrix}^T$ the channel value Φ_{kl} at each grid point $\begin{bmatrix} k\ l \end{bmatrix}^T$ is obtained by application of a Gaussian kernel function:

$$\Phi_{kl}(u, v, r) = r\mathrm{g}(d, \sigma) = re^{-0.5(d/\sigma)^2} \quad \text{for} \quad d^2 = (u-k)^2 + (v-l)^2 \qquad (2)$$

We only discuss the fully isotropic kernel here. Such an encoding of 2D variables is realised more efficiently as the outer product of the 1D channel vectors for the two components, u and v.

For several points the combined channel representation is simply given by the averaged channel matrix. An example with four thus encoded points is given in figure 1a and 1b for different kernel widths σ. Observe that there is an interference between two points if they are too close to each other. The number of channels, their distance and the standard deviation of the used kernel thus limits how many points we can represent simultaneously.

[3] If no confidence is available, we simply set $r = 1$.

2.2 Encoding of Lines

Often an image measurement does not give the exact location of the parameter we want to estimate, but only determines that it lies somewhere in a one-dimensional subspace. We assume that such a linear constraint is given either in standard, or in normalised form:

$$a\,x + b\,y + c = 0 \quad \text{or} \quad x\cos\phi + y\sin\phi - \rho = 0\,, \tag{3}$$

with $[\cos\phi,\ \sin\phi,\ -\rho] = \frac{1}{\sqrt{a^2+b^2}}\,\big[a\ b\ c\big]$. All points (x, y) which satisfy (3) lie on the line. The distance of a specific grid point (k, l) to the line is then given by $d = |k\cos\phi + l\sin\phi - \rho|$. Channel values are again obtained by applying the Gaussian kernel to the distance:

$$\Phi_{k,l}(\rho, \phi, r) = r\mathrm{g}(d, \sigma) = re^{-0.5(d/\sigma)^2} \quad \text{for} \quad d^2 = k\cos\phi + l\sin\phi - \rho\,. \tag{4}$$

Each channel value encodes the likelihood that the motion has the value of the corresponding grid point. An example with four thus encoded lines is given in figure 1c and 1d for different σ.

2.3 Point Decoding

In the decoding step we want to extract the position of local peaks in the channel matrix. First we determine local maxima with grid point accuracy at (k, l). Then we model the channel values in a small neighbourhood (e.g. 3×3 or 5×5) around the local maximum using a 2D Gaussian with centre position $\mathbf{s}$, amplitude r and covariance matrix $\mathbf{C}$:

$$g(\mathbf{p} - \mathbf{s}, r, \mathbf{C}) = r\ \exp\big(-0.5(\mathbf{p} - \mathbf{s})^T\mathbf{C}^{-1}(\mathbf{p} - \mathbf{s})\big) \tag{5}$$

where $\mathbf{p} = \big[x\ y\big]^T$ denotes local grid point coordinates. We can express the covariance matrix and its inverse explicitly as:

$$\mathbf{C} = \begin{bmatrix}\sigma_x^2 & \sigma_{xy}\\ \sigma_{xy} & \sigma_y^2\end{bmatrix} \quad \text{and} \quad \mathbf{C}^{-1} = \frac{1}{\sigma_x^2\sigma_y^2 - \sigma_{xy}^2}\begin{bmatrix}\sigma_y^2 & -\sigma_{xy}\\ -\sigma_{xy} & \sigma_x^2\end{bmatrix}\,. \tag{6}$$

For each point in the decoding neighbourhood we thus obtain one constraint $\Phi_{\mathbf{p}} = g(\mathbf{p} - \mathbf{s})$. After taking the logarithm this constraint becomes:

$$\ln g(\mathbf{p} - \mathbf{s}) = \ln r - \frac{(x - s_x)^2\sigma_y^2 - 2(x - s_x)(y - s_y)\sigma_{xy} + (y - s_y)^2\sigma_x^2}{2(\sigma_2^2\sigma_y^2 - \sigma_{xy}^2)}\,. \tag{7}$$

This can be written as the scalar product between a known vector $\mathbf{a}$ and an unknown parameter vector $\mathbf{m}$ with:

$$\mathbf{a} = 0.5 \begin{bmatrix} 1 \; 2x \; 2y \; -x^2 \; -y^2 \; -2xy \end{bmatrix}^T , \tag{8}$$

$$\mathbf{m} = \frac{1}{\sigma_x^2\sigma_y^2 - \sigma_{xy}^2} \begin{bmatrix} 2\ln r(\sigma_x^2\sigma_y^2 - \sigma_{xy}^2) - s_x^2\sigma_y^2 + 2s_x s_y \sigma_{xy} - s_y^2\sigma_x^2 \\ s_x\sigma_y^2 - s_y\sigma_{xy} \\ s_y\sigma_x^2 - s_x\sigma_{xy} \\ \sigma_y^2 \\ \sigma_x^2 \\ -\sigma_{xy} \end{bmatrix} \tag{9}$$

Stacking the constraints for each pixel on top of each other we obtain a least-squares system: $\mathbf{A}\mathbf{m} = \ln \boldsymbol{\Phi}$. The solution is obtained using the pseudo-inverse:

$$\mathbf{m} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \ln \boldsymbol{\Phi} . \tag{10}$$

We recognise the inverse covariance as:

$$\tilde{\mathbf{C}}^{-1} = \begin{bmatrix} m_4 \; m_6 \\ m_6 \; m_5 \end{bmatrix} \quad \text{thus} \quad \tilde{\mathbf{C}} = \frac{1}{m_4 m_5 - m_6^2} \begin{bmatrix} m_5 & -m_6 \\ -m_6 & m_4 \end{bmatrix} . \tag{11}$$

From (9) we find the position and peak amplitude to be given by:

$$\begin{bmatrix} \tilde{s}_x \; \tilde{s}_y \end{bmatrix}^T = \tilde{\mathbf{C}} \begin{bmatrix} m_2 \; m_3 \end{bmatrix}^T \; ; \; \tilde{r} = \exp(0.5(m_1 + m_4 s_x^2 + m_5 s_y^2 + 2m_6 s_x s_y)) . \tag{12}$$

The final solution is obtained by adding the centre grid point offset $\begin{bmatrix} \tilde{u} \; \tilde{v} \end{bmatrix} = \begin{bmatrix} k \; l \end{bmatrix} + \begin{bmatrix} \tilde{s}_x \; \tilde{s}_y \end{bmatrix}$.

The expectation of the estimated covariance matrix is the sum of the noise covariance $\tilde{\mathbf{C}}_n$ and that of the encoding kernel $\mathbf{C}_b = \text{diag}(\sigma^2, \sigma^2)$. Hence we can compute the covariance matrix of our estimated result to be: $\tilde{\mathbf{C}}_n = \tilde{\mathbf{C}} - \mathbf{C}_b$. Also note that the estimated amplitude $\tilde{r}$ encodes the peak likelihood and thus directly serves as a certainty measure.

In order to determine to what extent the aperture problem persists for the considered solution we can define the quotient of the eigenvalues of the covariance matrix as a simple measure. Let $\lambda_1 \geq \lambda_2$ be the eigenvalues of $\tilde{\mathbf{C}}$, then we define:

$$r_a = \lambda_2 / \lambda_1 . \tag{13}$$

To summarise: for each local peak in the channel matrix, the decoding extracts the mode location $(\tilde{u}, \tilde{v})$, the amplitude $\tilde{r}$, the covariance $\tilde{\mathbf{C}}$, and an aperture measure r_a.

2.4 Multiple decodings

The above described decoding scheme tends to give several similar solutions if the local channel matrix structure is elongated, i.e. if r_a from (13) is small. Thus we also do a postprocessing which removes multiple solutions. As a first step, we disregard solutions with a Mahalanobis distance larger than 1 from the initial grid point, i.e.

$$\left[\tilde{s}_x\ \tilde{s}_y\right] \tilde{\mathbf{C}}^{-1} \left[\tilde{s}_x\ \tilde{s}_y\right]^T < 1\,. \tag{14}$$

Additionally, we check if multiple solutions are within each other's Mahalanobis distance, and if so, we keep the one which has the largest aperture measure r_a.

The thus estimated peak locations and associated covariance matrices are shown in figure 1. For isolated points the covariance vanishes, i.e. we have perfect reconstruction. Increasing the kernel size σ leads to stronger interference as can be seen in figure 1b. However the elongated shape of the covariance matrix correctly captures this interference. For linear constraints we find that the intersections are correctly found, see figure 1c. Observe that the angle between the lines determines the covariance in the reconstructed point, for 90° we should have an isotropic covariance. However for small values of σ we find a slight anisotropy caused by quantisation effects. For larger σ (figure 1d) this effect disappears.

2.5 Line Decoding

In cases where the point decoding fails, i.e. when the decoded covariance matrix has a zero, or negative determinant, we revert to decoding a line instead. We do this by constraining the inverse covariance matrix to be singular. The singularity is enforced by eigenvalue decomposition on $\tilde{\mathbf{C}}^{-1}$, and setting the smallest eigenvalue to zero.

$$\begin{bmatrix} m_4 & m_6 \\ m_6 & m_5 \end{bmatrix} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T \ \Rightarrow\ \tilde{\mathbf{C}}^{-1} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T = \begin{bmatrix} \tau_x^2 & \tau_x \tau_y \\ \tau_x \tau_y & \tau_y^2 \end{bmatrix}\,. \tag{15}$$

For a singular matrix the peak position $\left[\tilde{s}_x\ \tilde{s}_y\right]^T$ is not well defined. Instead we compute the minimum norm solution, cf. (12), of $\tilde{\mathbf{C}}^{-1}\left[\tilde{s}_x\ \tilde{s}_y\right]^T = \left[m_2\ m_3\right]^T$:

$$\begin{bmatrix} \tilde{s}_x \\ \tilde{s}_y \end{bmatrix} = \frac{(\tau_x m_2 + \tau_y m_3)}{\tau_x^2 + \tau_y^2} \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix}\,. \tag{16}$$

Finally, the covariance is approximated by:

$$\tilde{\mathbf{C}} = 1/\lambda_1 (\mathbf{e}_1 \mathbf{e}_1^T + 10\,000 \mathbf{e}_2 \mathbf{e}_2^T)\,. \tag{17}$$

Note that the factor 10 000 is quite arbitrary. It is just a convenient approximation of the infinity, such that the accuracy of the orientation of $\tilde{\mathbf{C}}$ is retained.

3 Optical Flow

We now apply the presented framework to the computation of image motion. From the assumption of conserved intensity the standard optical flow constraint equation is obtained as:

$$u f_x(x,y) + v f_y(x,y) + f_t(x,y) = 0\,. \tag{18}$$

Here f_x, f_y, and f_t denote the signal derivatives along space and time dimensions, and $[u, v]^T$ the motion. As there is only one equation with two unknowns, the solution is constrained to lie on a line in the parameter space. This inherent ambiguity is often referred to as the aperture problem. We encode this linear constraint as described in section 2.2, and obtain a blurred line constraint at each spatial position.

To obtain a unique solution some form of spatio-temporal smoothness is usually required. Here we simply assume the motion in each layer to be constant in a spatial neighbourhood. The channel matrix for such a neighbourhood is then obtained by averaging the individual matrices. Instead of a standard average it is desirable to give more weight to the central pixel. This is readily achieved by the use of an averaging filter $g(x, y)$ such as a Gaussian or binomial. Furthermore we might want to utilise a certainty $w(x, y)$ at each pixel. The gradient magnitude is a possible choice. In any way this certainty will be zero outside the image thus reducing border effects. The integrated channel matrix is given by a normalised average:

$$\Phi'_{kl} = \frac{g * (w \cdot \Phi_{kl})}{g * w} \tag{19}$$

where $*$ denotes convolution. We use the well known "Hamburg taxi" sequence (figure 2) to demonstrate the algorithm. Two thus computed channel matrices are shown in figure 2b and 2c for the locations indicated in figure 2a. Note that the averaged channel matrix corresponds to a sampled likelihood function with:

$$p(u, v|f) \simeq \sum_{x,y} g(x, y)w(x, y) \exp\left(-\frac{(u\, f_x + v\, f_y + f_t)^2}{2\sigma^2(f_x^2 + f_y^2)}\right) \ . \tag{20}$$

(All derivatives above have an implicit spatial coordinate argument.) This can easily be derived from (3) and (4). Compare this likelihood to the standard least-squares likelihood function [13]:

$$p(u, v|f) \simeq \exp\left(\frac{-1}{2\sigma^2} \sum_{x,y} g(x, y)w(x, y)(u\, f_x + v\, f_y + f_t)^2\right) \ . \tag{21}$$

These two likelihoods are illustrated in figure 2c and 2d for an area where a moving car is occluded by a tree. We observe that (20) clearly distinguishes the two motions while (21) averages them. The summation in (20) can be thought of as a voting mechanism which makes the approach very robust to outliers, similar to a generalised Hough transform. Note that when there is only one solution and no outliers the expectation values of (20) and (21) coincide.

Points where more than one solution is obtained are indicated in figure 2e. The outlines of two cars are clearly visible. There are no multiple motions around the bright car as its slow movement can not be separated from that of the background in this case ($\sigma = 0.2\Delta u$). The certainty of the dominant estimate is shown in figure 2f, this also drops slightly around the brighter car. Finally we show the first and second estimated motion in figure 2g and 2h respectively.

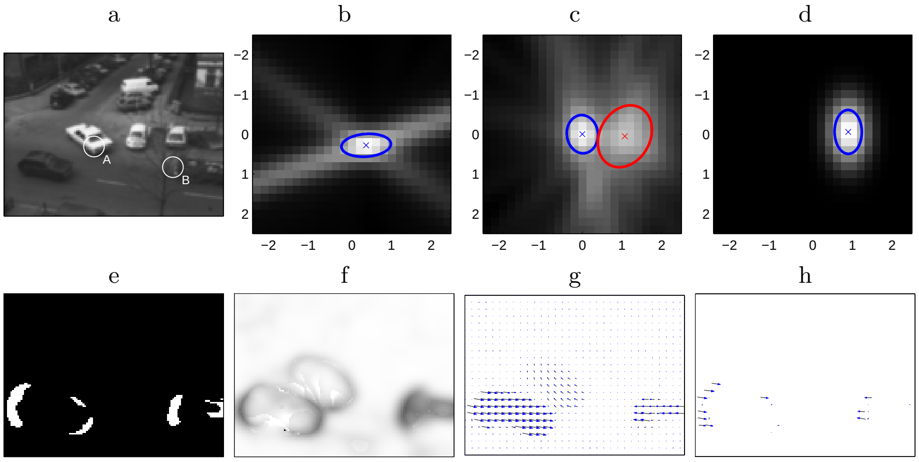

Fig. 2. Optical flow example. a) frame 25 with marked positions A and B. b) and c) show channel matrices at positions A and B respectively. d) gives the LS likelihood (21) at point B. e) shows the pixels where more than one solution is obtained, f) shows the confidence in the first estimate and g) and h) show the first and second solutions as vector plots.

Around the cars their movements are captured in the less dominant second solution.

It is possible to extract multiple motions at motion discontinuities. This is illustrated on a synthetic sequence where all four quadrants move in different directions, an example frame is given in figure 3a. The number of solutions is shown in figure 3b; At the centre we get up to four estimates and at the other discontinuities we obtain two solutions. The vector plot (figure 3c) illustrates that the motions are correctly estimated. The amplitude of the dominant peak drops near the discontinuities as the energy is distributed to several peaks, see 3d.

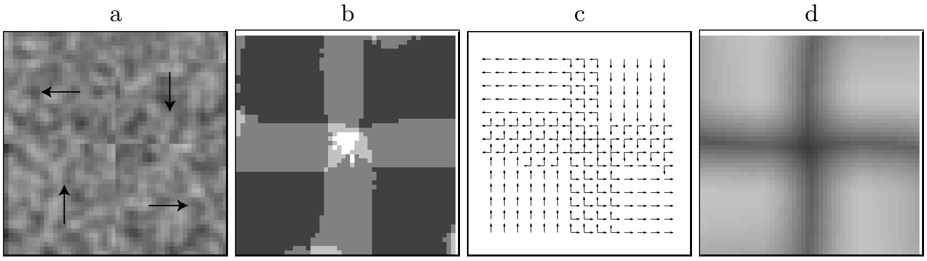

Fig. 3. Multiple motions at discontinuities. a) example image, b) number of solutions (range [0 4]), c) vector plot and d) confidence (peak amplitude) in the first estimate.

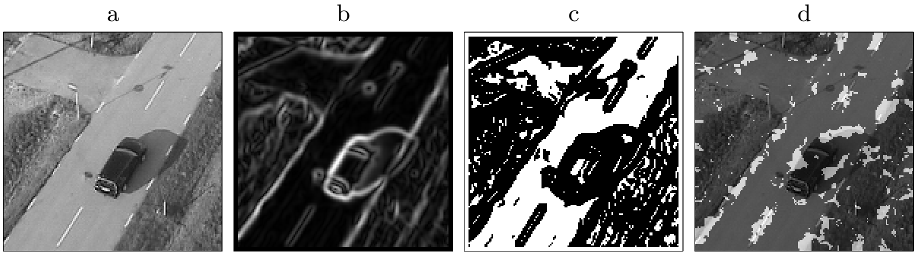

Fig. 4. Helicopter sequence. Left to right: input frame, Gradient magnitude, Thresholded gradient magnitude, thresholded r_a map (overlaid on image).

4 Scale Selection for Optical Flow Estimation

The “Hamburg taxi” sequence is quite convenient in the sense that it does not contain elongated structures, and thus no serious aperture problems exist. In general however we may need to integrate information over quite a large region in order to get rid of aperture problems. If a local region doesn’t suffer from aperture problems, however it is not desirable to use a too large estimation region. In fact, increasing the estimation region reduces the spatial localisation accuracy of the estimated motion. This is known as *the uncertainty principle* [2]. In order to keep the spatial localisation accuracy, and at the same time adapt the size of the estimation region, we compute a low-pass pyramid for each channel.

Figure 4a shows a frame from a more difficult sequence. In this sequence, the car is moving forward, and the camera is translating upwards and to the left, to compensate for the car motion. Figure 4b shows the gradient magnitude $f_m = \sqrt{f_x^2 + f_y^2}$, and figure 4c indicates regions where the magnitude is so small ($f_m < 0.005$) that the BCCE constraints are unreliable due to low SNR. For such regions, we don’t provide any constraint at all, instead we set the channel matrix to all zeros. Figure 4d indicates regions where an optical flow estimation according to section 3 (using a 21-tap binomial filter to average the channel matrices) resulted in a dominant motion with a very low aperture measure ($r_a < 0.01$).

Both types of problem regions (those indicated in figure 4c and 4d) have to be dealt with in some way. For the regions in figure 4c we could either extrapolate motion estimates from neighbouring regions, as is typically done in dense optical flow techniques, or we could leave the motion undefined, and leave it to a yet-to-be-specified post-processing algorithm to infer the correct motion. For the regions indicated in figure 4d we have valid BCCE constraints, but we have failed to resolve the aperture problem, and thus need to perform the estimation in a larger neighbourhood. This is the topic of this section.

First we generate a pyramid as follows:

1. Encode the BCCE constraints as channel matrices in each pixel.

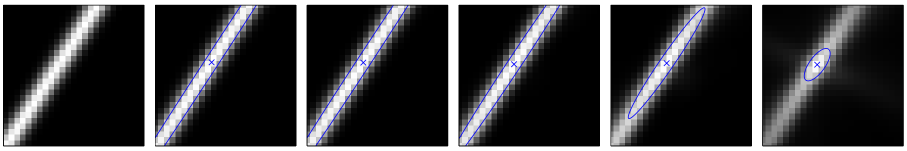

Fig. 5. Example of channel matrix behaviour under blurring. Left: initial constraint line. Second to sixth images show channel matrices from successive scales in the pyramid.

2. Perform an initial spatial average, e.g. with a 15-tap binomial filter.
3. Average again using a 4-tap binomial filter.
4. Subsample to obtain the next coarser scale in the pyramid.
5. Repeat 3 and 4 until sufficiently many scales are obtained.

Figure 5 demonstrates how channel matrices at a motion discontinuity behave under blurring. Here we can see how the aperture problem is dealt with at increasingly coarser scales. The first three scales all produce very elongated solutions, indicating aperture problem or near aperture problem uncertainty. the dominant modes at the fourth and fifth scales however have more concentrated covariance matrices, and are thus good descriptions of the motion in the region. This example motivates the following scale selection algorithm:

1. Decode at finest scale
2. For all decodings with an aperture measure below a given threshold:
3. Replace with decoding at coarser scale if within Mahalanobis distance, and better wrt. aperture measure.
4. Go back to 2.

This algorithm is demonstrated in figure 6. Here we can see that initial estimates made in a relatively small region (15-tap binomial) can be improved by replacing estimates with low aperture measures. However, the method fails on very elongated structures, such as e.g. the road line at the bottom of the image. The reason for this is that the number of votes for the same motion constraint is so high, that the estimates at the end of the line fails to change the shape of the covariance matrix significantly. This clearly indicates that a better aperture measure would be desirable.

5 Experiments

We now demonstrate the difference in behaviour between the described algorithm and the least-squares optical flow algorithm of Lucas and Kanade [14]. We compute optical flow by solving a system of BCCEs in each local neighbour-

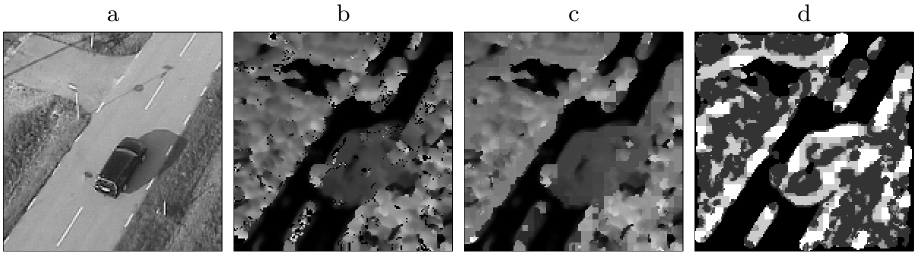

Fig. 6. Estimated motion. a) Central input frame b) Magnitude of motion estimate at scale 1 c) Magnitude of motion estimate after integrating scales 1–5. d) Chosen scale (brighter means higher scale, black means no estimate).

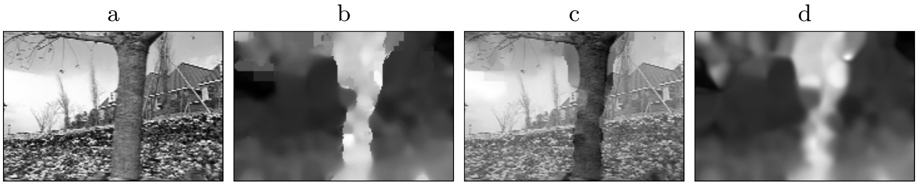

Fig. 7. Comparison with linear optical flow on the "flower garden" sequence. a) centre input frame (No. 5), b) magnitude of estimated motion, c) magnitude overlaid on input frame, d) result from least-squares method.

hood:

$$\mathbf{W}\underbrace{\begin{bmatrix}| & |\\ f_x & f_y\\ | & |\end{bmatrix}}_{\mathbf{A}}\begin{bmatrix}u\\ v\end{bmatrix} = \mathbf{W}\underbrace{\begin{bmatrix}|\\ f_t\\ |\end{bmatrix}}_{\mathbf{b}} \quad\Rightarrow\quad \begin{bmatrix}u\\ v\end{bmatrix} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{b}\,. \tag{22}$$

Here **W** is a diagonal matrix containing the spatial weights in the neighbourhood.

Figure 7 shows a comparison of the channel matrix method and the least-squares flow method (22). Figure 7a shows the centre input frame, and in 7b we have depicted the magnitude (mainly contains the horizontal component) of the estimated motion field using the channel matrix method. We used 3 × 3 Sobel derivative filters, but blurred the input frames using a 9-tap binomial to increase the spatial support of the derivatives. We have used a channel representation with 25×25 channels with $\mathit{\Delta} u = 0.35$, and $\sigma = 1.3\mathit{\Delta} u$. The initial smoothing was done using a 61-tap binomial filter, and the rest of the pyramid was built using 4-tap filters. The motion magnitude is overlaid on the central input frame in figure 7c, to illustrate the localisation of the result. As can be seen, the localisation of the tree edges are quite good when there is structure behind the tree. Higher up, however we see that the motion of the tree spills over onto the background as well.

The result for the least-squares flow method is shown in figure 7d using the same derivative filter responses as input. Again we used a spatial neighbourhood of a 61-tap binomial filter to solve for (u, v). As can be seen, the edges are significantly more blurred, and the least-squares flow gives erroneous motions at the same places as the channel matrix, (see e.g. the bright patches at the bottom right of the image, and on the upper right part of the tree trunk). This indicates that these errors are due to erroneous BCCEs, and not to the motion estimation step. By using better derivative filters, e.g. Sharr filters [15], or by switching to a different motion constraint estimation we should thus be able to improve the results.

6 Concluding remarks

We have presented a framework for encoding local motion constraints in a representation where averaging yields robust estimation. We wish to emphasise that the main purpose of this paper was to demonstrate the framework, and not to suggest a final motion estimation method. The channel representation framework allows the BCCE constraint to be replaced, e.g. by a phase based [16], or 3D-orientation constraint. Furthermore, we can easily combine both measurements that yield motion constraints and full motion estimates (as obtained from e.g. feature matching methods) by encoding them using the line encoding and the point encoding respectively.

For integration of estimates at different scales, the results are somewhat disappointing, since only minor improvements compared to the single scale method are obtained. Better results could probably be obtained by instead letting the covariance of the initial estimates guide the shape of the estimation region at coarser scales.

7 Acknowledgements

The work presented in this paper was supported by sixth framework programme COSPAL (COgnitive Systems using Perception–Action Learning), and Swedish foundation for strategic research project VISCOS (Vision in Cognitive Systems).

References

1. Jepson, A., Black, M.: Mixture models for optical flow. Technical Report RBCV-TR-93-44, Res. in Biol. and Comp. Vision, Dept. of Comp. Sci., Univ. of Toronto (1993)
2. Granlund, G.H., Knutsson, H.: Signal Processing for Computer Vision. Kluwer Academic Publishers (1995) ISBN 0-7923-9530-1.
3. Ayer, S., Sawhney, H.S.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In: Proceedings of the fifth International Conference on Computer Vision. (1995)
4. Spies, H., Forssén, P.E.: Two-dimensional channel representation for multiple velocities. In: Proceedings of the 13th Scandinavian Conference on Image Analysis. LNCS 2749, Gothenburg, Sweden (2003) 356–362

5. Shizawa, M., Mase, K.: Simultaneous multiple optical flow estimation. In: ICPR, Atlantic City, NJ, USA (1990) 274–278
6. Nestares, O., Navarro, R.: Probabilistic estimation of optical flow in multiple band-pass directional channels. Image and Vision Comp. **19** (2001) 339–351
7. Mota, C., Stuke, I., Barth, E.: Analytic solutions for multiple motions. In: ICIP. Volume 2., Thessaloniki, Greece (2001) 917–920
8. Andersson, K., Knutsson, H.: Multiple hierarchical motion estimation. In: SPPRA, Crete, Greece (2002) 80–85
9. Beauchemin, S.S., Barron, J.L.: On the fourier properties of discontinuous motion. Journal of Mathematical Imaging and Vision **13** (2000) 155–172
10. Forssén, P.E.: Low and Medium Level Vision using Channel Representations. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden (2004) Dissertation No. 858, ISBN 91-7373-876-X.
11. Granlund, G.H.: An associative perception-action structure using a localized space variant information representation. In: Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC), Kiel, Germany (2000)
12. Nordberg, K., Granlund, G., Knutsson, H.: Representation and Learning of Invariance. In: Proceedings of IEEE International Conference on Image Processing, Austin, Texas, IEEE (1994) Also as Technical Report LiTH-ISY-R-1552.
13. Weiss, Y., Fleet, D.: Velocity likelihoods in biological and machine vision. In Rao, R. P. N., Olshausen, B. A., Lewicki, M. S., eds.: Probabilistic Models of the Brain: Perception and Neural Function. MIT Press (2001) 81–100
14. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: DARPA Image Understanding Workshop. (1981) 121–130
15. Weickert, J., Scharr, H.: A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. J. Visual Communication and Image Representation **13** (2002) 103–118
16. Felsberg, M.: Optical flow estimation from monogenic phase. In: IWCM04, Reisensburg, Germany (2004)