Robust motion estimation using spatial Gabor-like filters

E. Bruno\textsuperscript{a}, D. Pellerin\textsuperscript{a,b, *} 

\textsuperscript{a}Laboratoire des Images et des Signaux (LIS), INPG, 46 Av. Félix Viallet, 38031 Grenoble Cedex, France  
\textsuperscript{b}ISTG, Université Joseph Fourier, Grenoble, France

Received 11 March 2001; accepted 19 October 2001

Abstract

This paper presents a new algorithm for motion estimation. It combines Gabor-like filter decomposition and robust least-squares estimation in a multiresolution framework. The spatial Gabor-like filter bank, based on recursive implementation, provides a fast multichannel decomposition of frame sequence. Then, applying the brightness constancy constraint on each channel between two consecutive frames, an over-determined system of velocity equations at each pixel is obtained. In order to be robust to outliers, this over-determined system is solved using a robust least-squares technique. A multiresolution framework is used in order to manage large and small displacements. Two kinds of recursive filter implementation have been tested: whereas third order filtering is very similar to a real Gabor filter, first order recursive filters are fastest and can be implemented with very large scale integration (VLSI) analog circuit. Performances of our algorithm for the two filter implementations are tested on synthetic and real sequences, and are compared with other techniques. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Motion estimation; Gabor filter bank; Recursive filters; Robust least-squares

1. Introduction

Optical flow estimation is a fundamental problem in low level vision. It is the projection of the 3D velocity on the image plane and can be extracted from the image brightness variation. Different approaches for optical flow recovery have been proposed. These can be grouped into correlation, energy, phase (reputed to give accurate estimation) and differential approaches. The latter are based on the optical flow constraint equation [6]:

\[
\nabla I(x, y, t) \cdot V(x, y, t) + \frac{\partial I(x, y, t)}{\partial t} = 0
\]

where \( V(x, y, t) = (u_x, y, t, v_x, y, t) \) is the velocity and \( I(x, y, t) \) is the spatiotemporal luminance. Equation (1) has a low computational cost, but only provides the velocity normal component (aperture problem). To overcome this problem, at position \((x, y, t)\) more than one equation is needed to obtain the two velocity components.

An approach to constraint equation (1) is the multichannel strategy [16,18,2,12,14]. It consists in filtering an image sequence by a set of spatial or spatio-temporal filters and applying the optical flow constraint equation on each channel. Thus, this technique provides an over-determined system of equations (1). Multichannel decomposition finds additional information in the close spatial or spatio-temporal neighborhood and thus needs velocity constancy assumption over this neighborhood. Many of these techniques provide reliable optical flow. In
particular, Weber and Malik [18] have developed an algorithm with accurate and reliable performances, comparable to phase-based approaches [5]. They have used a set of spatio-temporal Gaussian derivative filters at a number of orientations and scales. However, this approach suffers from a heavy computational cost due to the spatio-temporal filtering. On the other hand, Bernard has recently proposed an attractive approach based on wavelet decomposition [2]. Using the Fast Wavelet Transform, the algorithm has a low computational complexity, but without outperforming Weber’s algorithm in terms of optical flow quality.

In this paper, a new algorithm using Gabor-like filters is proposed. These filters, based on recursive implementation of Gaussian filters, are fast and scale-independent. An accurate optical flow is then estimated from the Gabor-like filter output, using robust least-squares in a multiresolution framework. The next section describes in detail the multichannel strategy for motion estimation, and exposes our motivation for using Gabor-like filters. Sections 3 presents the first and third order recursive filters used in this paper, and the problems with filter bank parameter setting. Sections 4 and 5 describe the $M$-estimators and the multiresolution framework. These two techniques are a good way to enhance optical flow reliability. Optical flow estimated from different sequences is presented in Section 6.

2. Optical flow from multichannel image decomposition

In order to estimate the velocity vector at each pixel location with Eq. (1), one needs to add some constraints to the velocity field. Usually, a smooth assumption is assumed, which means that the velocity field is locally constant. The multichannel approach is based on this assumption. Let us consider the projection of the motion equation (1) onto a filter bank $G_i(x, y)$ (with $i = 1, \ldots, N$, and $N$ the number of filters):

$$\int \int \left( u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \right) \times G_i(x - x_0, y - y_0) \, dx \, dy = 0. \tag{2}$$

Then, if $(u, v)$ are constant under the spatial extent of $G_i(x, y)$:

$$u \frac{\partial I}{\partial x} * G_i + v \frac{\partial I}{\partial y} * G_i + \frac{\partial I}{\partial t} * G_i = 0 \tag{3}$$

* denotes the convolution operator. Using the commutativity property of the convolution product it follows:

$$u \frac{\partial (I * G_i)}{\partial x} + v \frac{\partial (I * G_i)}{\partial y} + \frac{\partial (I * G_i)}{\partial t} = 0. \tag{4}$$

The derivative operator inversion allows only the image sequence to be convolved by the filter bank $G_i$ instead of each spatio-temporal derivative. Finally, at a given position $(x, y, t)$, each filter output satisfies Eq. (4) which results in an over-constrained system:

$$\begin{bmatrix} \Omega^u_1 & \Omega^v_1 \\ \Omega^u_2 & \Omega^v_2 \\ \vdots & \vdots \\ \Omega^u_N & \Omega^v_N \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \Omega^u \end{bmatrix}, \tag{5}$$

where $\Omega^p = \partial (I * G_i)/\partial p$.

In order to solve the system (5), two equations at least are needed. However, a stable solution of the system (5) is obtained when it is over-constrained and properly conditioned. In addition, to respect local velocity smoothness assumption, the spatial extent of $G_i$ cannot be too large. These conditions imply the use of filter banks with the smallest spatial extent possible, and which provide a set of subbands from the image sequence which are as independent as possible (i.e. filters which minimize the Heisenberg incertitude).

Several authors have proposed similar approaches using different filter banks: Weber and Malik [18] use a set of derivative Gaussian filters, Bernard’s method [2] is based on the fast wavelet transform (FWT). We can cite many other authors, such as Tsao and Chen [16] (Gabor filters), Simoncelli [14] (Gaussian filters) or Mitiche et al. [12] (many spatial operators). We can draw many lessons from these approaches. First, Weber and Malik have shown that multichannel strategy produces more constraint than region approaches. Secondly, in order to design filter banks, Bernard shows that complex filters provide more stable velocity estimation than real filters. Third, as filter bank convolution is time-consuming, one needs to find a fast implementation of it (such as Bernard’s algorithm...
The Gabor filter bank, used by Tsao and Chen, has several advantages in regard to our problem:

- it is a complex filter,
- it minimizes the Heisenberg incertitude,
- it allows scale, band-pass and orientation analysis of image.

In addition, Gabor filter banks are used for numerous early vision tasks, such as texture discrimination [19] or feature extraction [10]. They have also received support from neurophysiological experiments [8]. However, the computational load of the Gabor filtering stage is prohibitive. We get around this problem by using Gabor-like filters. Based on recursive implementation, these filters are very time efficient, and allow an intensive use of them.

3. Gabor and Gabor-like filter banks

The Gabor filter in the ith channel has an impulse response $G_i(x, y)$:

$$G_i(x, y) = \frac{1}{2\pi \sigma^2} e^{-\left(x^2+y^2\right)/2\sigma^2} e^{2\pi i f_0(x \cos \theta_i + y \sin \theta_i)}$$

(6)

where $f_0$, $\theta_i$ and $\sigma$ are, respectively, the filter central frequency, orientation and scale. The direct implementation (convolution) of 2D Gabor filters is a very lengthy stage. There exists a more efficient recursive implementation of these filters [15].

3.1. First and third order recursive Gabor-like filter

The output $y(k)$ of a 1D Gabor filter excited by an input signal $x(k)$ is written:

$$y(k) = x(k) \ast \left[ K e^{-\left(k^2/2\sigma^2\right)} e^{2\pi i f_0 k} \right]$$

(7)

by developing relation (7), it can be shown that:

$$y(k) = \left( [x(k) e^{-j2\pi f_0 k}] \ast K e^{-k^2/2\sigma^2} \right) e^{2\pi i f_0 k}.$$  

(8)

Thus, Gabor filter could be broken down into three successive stages: a modulation, a low-pass filtering (Gaussian envelope) and a demodulation. Gaussian filtering is carried out using a recursive filter. The number of operation per pixel using a recursive Gaussian filter implementation is constant whatever the filter selectivity. The computation time only depends on the order of the recursive filter.

We have tested 3rd and 1st order filters (Fig. 1). 3rd order filters have a small residual error when compared with a Gaussian filter. On the other hand, the 1st order filter is the fastest implementation, with a shape closer to a Lorenz function. Note that Shi [13]
has proposed an attractive analog VLSI implementation of Gabor-like filter banks. He described the design of cellular neural networks which compute the outputs of filters equivalent to the 1st order recursive filter we use. These filters, whose coefficients depend on selectivity, are expressed as a product between a causal and an anti-causal term:

\[ v[n] = ax[n] - \sum_{i=1}^{N} b_i v[n-i], \]

\[ y[n] = ax[n] - \sum_{i=1}^{N} b_i y[n+i]. \]

With \( x[n] \) the input signal and \( y[n] \) the output signal. The coefficients \( a \) and \( b_i \) are computed using formulae deduced by Vliet [17].

### 3.2. Gabor-like filter bank design

Gabor filter banks are usually isotropic and defined by a central frequency \( f_0 \), a scale \( \sigma \), and a number of filters \( N \) which determine the orientations \( \theta_l = \pi l / N \) (Fig. 2). These filter banks are then denoted \( G_{f_0,\sigma,N} \).

Filter bank design is a very crucial point in our problem. It is used to provide a multichannel image description where each channel must be as least redundant as possible in regards to the others. Let us suppose two Gabor filters (6) with the same scale \( \sigma \), the same central frequency \( f_0 \), and different orientations \( \theta_1 \) and \( \theta_2 \). The redundancy between these two subbands is

\[
R(\delta\theta) = \frac{\langle G_1, G_2 \rangle}{||G||^2} = \exp(-2\pi^2 f_0^2 \sigma^2 (1 - \cos(\delta\theta)))
\]

with \( \delta\theta = \theta_2 - \theta_1 \in [\pi/2, \pi/2] \).

So as to minimize \( R \), one needs to increase \( \sigma \) and \( f_0 \) and set \( N = 2 \) (\( \delta\theta = \pi/2 \)). Unfortunately, none of these conditions are compliant with our problem:

- The larger the scale \( \sigma \) is, the more the redundancy \( R \) decreases to 0. However, a larger \( \sigma \) supposes that motion is constant over large image regions (in fact, under the filter spatial support). This assumption fails when motion varies in these regions (affine motion, occlusion, boarder motion...). We have to use filters of the smallest possible support size but large enough to get around the aperture problem. Several authors [1,3] suggest that a 5 × 5 window is a lower bound to estimate a reliable velocity vector. Following this suggestion, typical interval for \( \sigma \) could be [2.0, 4.0].

- Central frequency \( f_0 \) should be as high as possible to minimize \( R \), but lower than the boundary imposed by spatial derivative estimations. We use the filter \([-1,8,0,-8,1] / 12 \) which provides a good derivative approximation for spatial frequency lower than 0.2. Thus, considering the frequency width of Gabor filters (\( \sigma = 3.0 \)), \( f_0 \) should be set to a value lower than

\[
f_0 \leq 0.2 - \frac{1}{2\pi\sigma} \approx 0.15.
\]

- The more we use Gabor filters, the more the redundancy \( R \) increases to 1. We need \( N \) greater than 2 to obtain an over-determined system (5). On the other hand, using many filters implies huge computational load, without providing anymore constraints on the system (5). Fig. 3 displays redundancy evolution when \( N \) grows, with \( \sigma = 3.0 \) and \( f_0 = 0.15 \). Considering that if \( R > 0.5 \), subbands become very correlated, around 5 or 6 filters is enough to over-constrain the system (5) and estimate motion.
To sum up, Gabor filter banks parameters can be set as follows:
\[
\sigma \in [2.0, 4.0], \quad f_0 < 0.15, \quad N \in [3, 6]. \quad (12)
\]
Fig. 4 shows a Gabor filter with parameters in the interval defined by (12).

Because Gabor filters are not null when frequencies go to zero, each filter output shares the DC component. This effect is stronger when the spectrum decreases in \(1/f\) (typical natural image spectrum) and results in increasing redundancy between Gabor output. Then, in order to reduce the low frequency contribution, images are prefiltered by a Difference Of Gaussian (DOG) filter. The scales of the two recursive Gaussian filters for the DOG are set to \(\sigma_1 = 2.0\) and \(\sigma_2 = 0\). Fig. 5 displays the redundancy evolution in function of \(\sigma\) when input signal is prefiltered by a DOG filter. The DOG prefiltering effectively reduces the redundancy when the scale tends towards zero.

We have seen that designing an optimal filter banks for motion estimation is a trade-off between many opposing constraints. According to the previous discussion, we can define intervals for the different parameters (12). However, final setting must be chosen empirically by experimental tests. This point is discussed later in Section 6.

4. Robust least-squares estimation

The problem now is to estimate the velocity field \(\mathbf{V} = (u, v)\) from the over-constrained system of Eq. (5). Since the Gabor filter outputs are complex, and the velocity is real valued, the system can be written as a real system (in matrix notation):
\[
\begin{bmatrix}
\text{Re}(M) \\
\text{Im}(M)
\end{bmatrix}
A
\begin{bmatrix}
V \\
Y
\end{bmatrix}
= 
\begin{bmatrix}
\text{Re}(B) \\
\text{Im}(B)
\end{bmatrix}. \quad (13)
\]

To solve system (13), one may use the least-squares technique. It provides an estimate \(\mathbf{V}\) by minimizing the following sum of squared errors \(r_i\):
\[
\mathbf{V} = \min_{u, v} \sum_i r_i^2 \quad (14)
\]
with \(r_i = uA_{i,1} + vA_{i,2} + Y_i\) and \(A_{i,j}\) is the element of the \(i\)th line and \(j\)th column of the matrix \(A\).

A robust estimation technique is needed, because outlying data could have bad results on the accuracy of the least-squares estimate. The \(M\)-estimators reduce the effect of outliers by replacing the squared residuals by another function of the residuals [7], then Eq. (14)
Fig. 4. Gabor filter frequency response with and without DOG prefiltering (with $f_0 = 0.15$, $\sigma = 3.0$, $\sigma_1 = 2.0$ and $\sigma_2 = 0.0$).

Fig. 5. Redundancy $R$ vs. scale $\sigma$ with $f_0 = 0.15$ and $N = 6$ for the 3rd order recursive filter with and without a DOG prefiltering.

becomes:

$$V = \min_{u,v} \sum_i \rho(r_i).$$  \hspace{1cm} (15)$$

This technique can be implemented as an iterated re-weighted least-squares:

$$V = \min_{u,v} \sum_i w_i r_i^2$$  \hspace{1cm} (16)
Fig. 6. (a) 3-level pyramidal decomposition and (b) the equivalent Gabor filter bank.

with \( w_i = \frac{1}{x}(\partial \rho (r_i) / \partial x) \) (\( r_i \) calculated by a first least-squares).

In order to suppress outliers from the minimisation (16), we use the Tukey’s biweight function which is a truncated \( \rho \) function:

\[
\rho(x) = \begin{cases} 
\frac{c^2}{\lambda}(1 - (1 - \frac{x}{\lambda})^2)^3 & \text{if } |x| \leq C, \\
\frac{c^2}{\lambda} & \text{if } |x| > C 
\end{cases}
\]  

and then:

\[
w(x) = \begin{cases} 
\left(\frac{x^2-c^2}{\lambda x}\right)^3 & \text{if } |x| \leq C, \\
0 & \text{if } |x| > C.
\end{cases}
\]  

Hence, if residual \( r_i \) is very small, the weight of data provided by the \( i \)th channel is near 1, otherwise it decreases as in the relation (18). If \( r_i \) is larger than \( C \), these data are removed from the estimation process. \( C \) is usually determined by computing a robust quantity [11]:

\[
z = \text{median}_i(|r_i - \text{median}_j(r_j)|)
\]  

such that \( C = \lambda z \) where \( \lambda \) is a proportionality factor which determines the robustness of the estimator.

To sum up, using matrix notations, robust least-squares estimation can be written as

\[
V = -(A^T W A)^{-1} A^T W Y
\]  

with

\[
W = \text{diag}(w_i).
\]

Singularity and round off errors in \( A^T W A \) inversion can be avoided by testing its eigenvalues: If \( (\lambda_2 / \lambda_1) < \epsilon \) (where \( \lambda_2 \) is the smallest eigenvalue) we reject the solution, and no velocity is computed at this location. Finally, a spatial median filter is applied to suppress aberrant solutions since the velocity field is assumed to be spatially smooth.

5. Coarse to fine scheme

The maximum estimated velocity is bounded by finite differences used to approximate derivatives in Eq. (1). Weber and Malik [18] have shown that for a sin wave signal of frequency \( f \), velocities allowed with an error less than 15% are \(|V| < 1/(2\pi f)\). Since we use band-pass filters, we can have an idea of the Gabor output spectrum. The upper frequency where Gabor filter is attenuated to a half is \( f_0 + 1/(2\pi\sigma) \). If we use this frequency in limiting the maximum displacement allowed, we find that for a Gabor filter bank \( G_{f_0,\sigma,N} \) the maximum velocity is

\[
|V| < \frac{\sigma}{2\pi \sigma f_0 + 1}.
\]  

This boundary can be pushed back by using a multiresolution strategy. A Gaussian low-pass pyramidal image decomposition (Fig. 6a) associated to the Gabor filter decomposition is equivalent to convolve the image sequence by a Gabor filter bank where
filter corona \( G_{2^{-i} f_0, 2^i \sigma, N} \) corresponds to the \( k \)th pyramid level (Fig. 6b). Then at levels \( k > 0 \), larger displacement can be estimated.

The coarse to fine scheme is as follows [4]:

At the highest level \( L \) of the pyramid, a coarse and large motion is estimated. This coarse motion is then transmitted to a fine level where a new estimation is performed. This process is repeated until the finest level is reached. Note that the multiresolution scheme reduces efficiently the computational load of the algorithm. Using this framework, the velocity field is:

\[
V = \sum_{k=0}^{L} 2^k V_k
\]

where \( V_k = (u_k, v_k) \) is the velocity at the level \( k \). The factor \( 2^k \) in expression (23) comes from the fact that the apparent velocity magnitude is reduced by a factor 2 each time images are sub-sampled by 2. \( V_k \) is estimated by applying the optical flow constraint equation at level \( k \):

\[
u_k \Omega^k_{k,i}(x, y) + v_k \Omega^k_{k,i}(x, y) + \tilde{\Omega}^i_{k,i}(x, y) = 0
\]

with \( \Omega^k_{k,i} = (((1 + G_{2^{-i} f_0, 2^i \sigma, j})/\tilde{\Omega}^i_{k,i}(x, y) \cdot \tilde{\Omega}^i_{k,i}(x, y)) \) is computed by warping frame \( I(t + 1) \) to \( I(t) \) with motion estimated in the higher level of pyramid:

\[
\tilde{\Omega}^i_{k,i}(x, y) = G_{2^{-i} f_0, 2^i \sigma, j} \ast

\left[ I \left( x - \sum_{j=k-1}^{L} 2^j u_j, y - \sum_{j=k-1}^{L} 2^j v_j, t + 1 \right) - I(x, y, t) \right].
\]

The warped frame is computed with a bilinear interpolation (because \( u_k, v_k \) are not integers).

6. Experimental results

We test our algorithm on both synthetic and real sequences. The angular error [1] is used to evaluate our estimation when the true motion field is known (in synthetic sequences downloaded from the Barron et al. FTP \(^1\)):

\[
z_v = \arccos \left( \frac{u u_r + v v_r + 1}{\sqrt{u^2 + v^2} \sqrt{u_r^2 + v_r^2} + 1} \right)
\]

with \((u, v)\) the estimated and \((u_r, v_r)\) the real optical flow. This provides a degree value which takes care of velocity modulus and orientation errors. We also highlight the density of pixels where motion is estimated.

6.1. Synthetic sequences

We have tested our algorithm (third order recursive filter) on Yosemite sequence (Fig. 7a and b). This sequence is used to set filter bank parameters because it has a complex enough motion to highlight different setting problems. The optical flow ranged in magnitude from zero at the focus to over 5 pixels per frame. To estimate such velocity magnitude, we used a three-level pyramid. In addition, there is an important border motion between sky and mountain, and motion estimation will be very sensitive to the scale magnitude of Gabor filters.

Fig. 8a displays angular error and vector density evolution in function of number of filters. Average and standard deviation errors are almost constant when \( N > 2 \), whereas density increases towards around 90% when \( N \geq 6 \). This is due to the robust least-square algorithm, which controls motion estimation errors: it rejects locations where information is too poor or too noisy (when a small number of Gabor filters is used) rather than computing false velocity vectors. Thus, choosing the number of filters is a trade-off between algorithm complexity and density of motion estimation.

Fig. 8b and c shows average and standard deviation errors as a function of scale \( \sigma \) and central frequency \( f_0 \). Best motion approximation occurs when scale and central frequency are around 3.0 and 0.14, respectively.

We have made the same studies using first order recursive filters. The settings are almost the same: \( f_0 = 0.14, \sigma = 5.0, \) and \( N \) is directly a function of velocity vectors density. Let us note that the scale is greater in this case, due to the shape of these filters.

Our algorithm has been tested on two other classical sequences [1], Translating Tree and Diverging Tree. Table 1 compares the average and standard deviation of the angular error of Weber’s algorithm [18], Fleet’s algorithm (phase based approach) [5], Bernard’s algorithm [2] and our algorithm using the first and third recursive Gabor filter bank. Weber et al. results were obtained by using 30 linear filters and 10 frames. The

\(^1\) ftp.csd.uwo.ca.
Fleet et al. algorithm ($\tau = 2.5$) used 46 3D convolutions and 21 frames (15 for Yosemite).

Third order recursive filter: The estimation error of our algorithm is better in two cases than the Weber’s algorithm. Particularly, for the Yosemite sequence the multiresolution strategy allows us to obtain a higher density flow field (Fig. 7c). However, our estimation is worse for the Translating tree, which has a constant translating motion in time. This can be explained by the fact that Weber’s algorithm needs 10 frames to recover a temporally smooth motion field, whereas our algorithm uses only 2 frames. In the same way, Fleet’s algorithm, which used 21 frames, outperforms for Translating and Diverging tree but gives worse results for Yosemite, with a lower density of vectors.

First order recursive filter: Compared to the Bernard, Fleet and Weber’s algorithm, our results remain satisfying. As planned, these filters do not provide a multichannel decomposition as independent as third order filters, and it results in a lower density of estimation and a greater angular error. However, in regards to the low computational complexity of such implementation, this solution is very attractive.
Table 1
Comparison of motion estimation errors for 3 image sequences. We use 6 complex Gabor filters \( f_0 = 0.14 \) and \( \sigma = 3.0 \) at each 3 pyramid levels and only 2 frames to estimate motion. The robust least-squares parameters \( \lambda \) and \( \varepsilon \) are set to \( 3 \) and \( 10^{-2} \), respectively.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Authors</th>
<th>Avg. error</th>
<th>Std. error</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yosemite</td>
<td>Weber and Malik</td>
<td>4.31°</td>
<td>8.66°</td>
<td>64.2%</td>
</tr>
<tr>
<td></td>
<td>Fleet and Jepson</td>
<td>4.25°</td>
<td>11.34°</td>
<td>34.1%</td>
</tr>
<tr>
<td></td>
<td>Bernard</td>
<td>6.50°</td>
<td>—</td>
<td>96.5%</td>
</tr>
<tr>
<td></td>
<td>3rd order filter</td>
<td>4.33°</td>
<td>7.95°</td>
<td>89.1%</td>
</tr>
<tr>
<td></td>
<td>1st order filter</td>
<td>5.13°</td>
<td>7.98°</td>
<td>81.6%</td>
</tr>
<tr>
<td>Diverging</td>
<td>Weber and Malik</td>
<td>3.18°</td>
<td>2.50°</td>
<td>88.6%</td>
</tr>
<tr>
<td>tree</td>
<td>Fleet and Jepson</td>
<td>0.99°</td>
<td>0.78°</td>
<td>61.0%</td>
</tr>
<tr>
<td></td>
<td>Bernard</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>3rd order filter</td>
<td>2.81°</td>
<td>1.72°</td>
<td>97.8%</td>
</tr>
<tr>
<td></td>
<td>1st order filter</td>
<td>2.82°</td>
<td>2.56°</td>
<td>92.3%</td>
</tr>
<tr>
<td>Translating</td>
<td>Weber and Malik</td>
<td>0.49°</td>
<td>0.35°</td>
<td>96.8%</td>
</tr>
<tr>
<td>tree</td>
<td>Fleet and Jepson</td>
<td>0.32°</td>
<td>0.38°</td>
<td>74.5%</td>
</tr>
<tr>
<td></td>
<td>Bernard</td>
<td>0.78°</td>
<td>—</td>
<td>99.3%</td>
</tr>
<tr>
<td></td>
<td>3rd order filter</td>
<td>0.83°</td>
<td>0.68°</td>
<td>94.3%</td>
</tr>
<tr>
<td></td>
<td>1st order filter</td>
<td>0.92°</td>
<td>0.60°</td>
<td>88.2%</td>
</tr>
</tbody>
</table>

6.2. Real sequences

We have tested our algorithm on three real sequences. First example is the well known Hamburg taxi sequence (Fig. 9). It is a street scene where four objects are moving: a taxi turning the corner, a car in the lower left driving from the left to the right, a van in the lower right, behind a tree, driving right to the left, and a pedestrian in the upper left. The optical flow of the three vehicles was well extracted, but the pedestrian was not detected. There is still some corrupted velocity vectors on the background, due to the noise present in this sequence.

A second test was made on a sequence acquired for lip segmentation and tracking (Fig. 10a) [9]. Dominant motion is due to mouth opening. Estimated optical flow (Fig. 10b) is close to perceptual view. We can observe a small asymmetrical mouth aperture (generally observed for human beings).

The last experience was made on Table Tennis sequence (Fig. 11a). Motion is due to camera displacement, player’s arms and ball motion. The optical flows...
Fig. 9. (a) Frame from Hamburg taxi sequence and estimated flow fields with (b) 3rd and (c) 1st order recursive filter banks.

Fig. 10. (a) Frame from speaker face sequence and estimated flow fields with (b) 3rd and (c) 1st order recursive filter banks.
estimated using 3rd and 1st order recursive filter banks are shown in Fig. 11b and c, respectively. Note that motions of fast objects, such as the small ball and the player’s arms, are well extracted.

7. Conclusion

Our algorithm provides accurate optical flow between only two frames in a robust multiresolution scheme. It is based on a spatial Gabor-like filter bank which allows the aperture problem to be avoided, then, M-estimators are used in order to estimate a robust velocity. A multiresolution scheme enables large and small motion to be recovered. Performances and time efficiency of our algorithm depend on filter approximation order (1st and 3rd order). In both cases, results are comparable to accurate techniques (Fleet et al. and Weber et al.) without needing spatiotemporal filtering over a large number of frames.

It is interesting to notice that Gabor filters are a common starting point in many low level vision applications. Such architecture could allow us in the future to merge motion with different indices extracted from Gabor filter banks.

References


