A Phase-Based Approach to the Estimation of the Optical Flow Field Using Spatial Filtering

Temujin Gautama and Marc M. Van Hulle, *Senior Member, IEEE*

***Abstract*—In this paper, we introduce a new technique for estimating the optical flow field, starting from image sequences. As suggested by Fleet and Jepson, we track contours of constant phase over time, since these are more robust to variations in lighting conditions and deviations from pure translation than contours of constant amplitude. Our phase-based approach proceeds in three stages. First, the image sequence is spatially filtered using a bank of quadrature pairs of Gabor filters, and the temporal phase gradient is computed, yielding estimates of the velocity component in directions orthogonal to the filter pairs' orientations. Second, a component velocity is rejected if the corresponding filter pair's phase information is not linear over a given time span. Third, the remaining component velocities at a single spatial location are combined and a recurrent neural network is used to derive the full velocity. We test our approach on several image sequences, both synthetic and realistic.**

***Index Terms*—Aperture problem, optical flow, phase-based, recurrent neural network.**

I. Introduction

MOTION is a prime source of information for determining the shape and structure of objects perceived in our environment. Therefore, accurate techniques for estimating the velocity field (optical flow field) are indispensable components of many vision applications. Horn and Schunck [14] were the first to develop a technique based on computing spatiotemporal differences from image sequences, which has spurred the development of a wide range of techniques and approaches for optical flow field estimation. In a review paper, Barron *et al.* [1], [3] evaluated nine different techniques, representative for various approaches, namely the differential, matching, energy-based, and phase-based ones. They have tested these algorithms on several standard image sequences, both synthetic, for which the *ground truth* motion fields are known, and realistic ones, for which the desired motion fields are not known and only qualitative comparisons can be made. One of their conclusions was that the phase-based technique of Fleet and Jepson [5] and the differential technique of Lucas and Kanade [18] produced the more accurate results overall. In another comparative study, Galvin *et al.* [9], [19] used more complex synthetic image sequences, for which the ground truth motion fields are available. Six out of the eight techniques that have been examined in this study, have also been used by Barron *et al.*, but leaving out the energy- and phase-based methods. Galvin *et al.* concluded that the technique developed by Lucas and Kanade [18] yields the best results.

Fleet and Jepson showed that the temporal evolution of contours of constant phase provides a better approximation to the local velocity than do contours of constant amplitude [5]–[7], [15]. They demonstrated that phase contours are more robust with respect to smooth shading and lighting variations, and more stable with respect to small deviations from image translations. They suggested tracking the constant phase contours by computing the phase gradient of (spatiotemporally) bandpassed images. They further showed that phase information can become unstable in the vicinity of phase singularities and have proposed a straightforward constraint that should be satisfied in order for the phase information to be reliable for the subsequent estimation of the two-dimensional (2-D) velocity.

We introduce a new phase-based approach to the estimation of the optical flow field, which is based on spatially filtering the images using a bank of quadrature pair filters, and not spatiotemporally as has been done by Fleet and Jepson [5]. This allows for the measurement of phase *nonlinearity*, rather than *instability*, on the basis of which a confidence measure can be defined that can be used to reject unreliable estimates. Every quadrature filter pair yields an estimate of the component of the velocity in the direction orthogonal to the filter pair's spatial orientation. Several such component velocities from a single spatial location are combined, and the 2-D velocity is found using a recurrent neural network. We test our technique on the image sequences used by Barron *et al.* [1], [3] and Galvin *et al.* [9], [19], such that results can be compared. Finally, we examine the influence of the free parameters that need to be chosen, and the technique's sensitivity to noise.

II. System Architecture

We estimate the optical flow field of an image sequence in three stages. First, we process the image sequence by spatially filtering the image at every time frame with a set of quadrature filter pairs and compute their phase responses. At every spatial location $\mathbf{x}(x_1, x_2)$, the temporal phase gradient for every filter $i, \phi_{t,i}(\mathbf{x})$, is computed, from which the *component* velocity, $\mathbf{v}_{c,i}$, is derived. Second, the reliability of these component velocities is examined and the unreliable ones are rejected. Third, at every spatial location, the component velocities from the different filter pairs are combined to produce an estimate of the *full* velocity, $\mathbf{v}$, at that position.

Manuscript received March 19, 2001; revised March 12, 2002. The work of T. Gautama was supported by a scholarship from the Flemish Regional Ministry of Education (GOA 2000/11). The work of M. M. Van Hulle was supported by research grants from the Fund for Scientific Research (G.0185.96N), the National Lottery of Belgium (9.0185.96), the Flemish Regional Ministry of Education, Belgium (GOA 95/99-06; 2000/11), the Flemish Ministry for Science and Technology (VIS/98/012), and the European Commission, Fifth Framework Programme (QLG3-CT-2000-30161 and IST-2001-32114).

The authors are with the Laboratorium voor Neuro- en Psychofysiologie, Katholieke Universiteit Leuven, B-3000 Leuven, Belgium (e-mail: temu@neuro.kuleuven.ac.be; marc@neuro.kuleuven.ac.be).

Publisher Item Identifier S 1045-9227(02)05568-6.

1045-9227/02$17.00 © 2002 IEEE

A. Component Velocity

We represent an image sequence by means of the outputs of a set of spatial filter pairs at every frame t. We use quadrature Gabor filter pairs, which are characterized by their center frequencies, (f_x, f_y), and the width of the enveloping (spatial) radially symmetric Gaussian, σ. We use filters with constant bandwidths of β octaves, measured at one standard deviation in the frequency domain, which results in a spatial width of

$$\sigma = \frac{2^\beta + 1}{(2^\beta - 1)2\pi\sqrt{f_x^2 + f_y^2}}.$$

The output of a quadrature Gabor filter pair is complex-valued, and we denote the phase component of the output by $\phi(\mathbf{x}, t)$.

Fleet and Jepson [5] have shown that the temporal evolution of contours of constant phase provides a good approximation to the motion field. Points $\mathbf{x}$ on such a contour satisfy $\phi(\mathbf{x}, t) = c$, where c is a constant, which, after differentiating with respect to t, yields

$$\nabla\phi(\mathbf{x}, t) \cdot \nabla\mathbf{x} = \nabla\phi(\mathbf{x}, t) \cdot (\mathbf{v}, 1) = (\phi_\mathbf{x}, \phi_t) \cdot (\mathbf{v}, 1) = 0$$

where $\nabla\phi(\mathbf{x}, t)$ is the temporal phase gradient vector and $\mathbf{v}$ is the full 2-D velocity. This allows us to write the temporal phase gradient as

$$\begin{aligned} \phi_t &= -(\mathbf{v} \cdot \phi_\mathbf{x}) \\ &= -\|\phi_\mathbf{x}\| (\mathbf{v} \cdot \phi_\mathbf{x}^n) \\ &= -\|\phi_\mathbf{x}\| \operatorname{proj}_{\phi_\mathbf{x}^n}(\mathbf{v}) \end{aligned} \tag{1}$$

where $\phi_\mathbf{x}^n$ denotes the normalized version of vector $\phi_\mathbf{x}$. Equation (1) can be interpreted as a formulation of the well-known aperture problem: the temporal phase gradient only yields information about the component of the full velocity in the direction of the spatial phase gradient $\phi_\mathbf{x}^n$. We denote this *component* velocity by $\mathbf{v}_c$, and compute it as

$$\mathbf{v}_c = \operatorname{proj}_{\phi_\mathbf{x}^n}(\mathbf{v})\phi_\mathbf{x}^n = \frac{-\phi_t(\mathbf{x})}{2\pi\left(f_x^2 + f_y^2\right)}(f_x, f_y). \tag{2}$$

Note that we substitute the spatial phase gradient $\phi_\mathbf{x}$ by the frequency vector $(2\pi f_x, 2\pi f_y)$ which, strictly speaking, only holds if the bandwidths of the filters approach zero.

For every filter pair i, the temporal phase gradient, $\phi_{t,i}(\mathbf{x})$ is computed from the temporal sequence of its phase components by performing a least-squares linear regression on the (t, ϕ)-pairs. We compensate for the phase wrap-around by adding or subtracting $(k \cdot 2\pi)$ if $|\phi(\mathbf{x}, t + \Delta t) - \phi(\mathbf{x}, t)|$ exceeds π ("phase-unwrapping"). Note that gradients larger than π cannot be accurately estimated this way and lead to large regression errors. The slope of the (t, ϕ)-regression line corresponds to the phase gradient, $\phi_t(\mathbf{x})$. We have opted to restrict ourselves to this simple method of phase-unwrapping, and to detect the inaccurate results in a second stage (see further); for more elaborate phase-unwrapping techniques, we refer to [10], [11], and [20].

We have tested this strategy on an (x, t)-case for a one-dimensional (1-D) quadrature Gabor filter pair with a wavelength of 12 pixels and a bandwidth of $\beta = 0.6$ octaves (the total operator length is set to $6\sigma = 55$ pixels). The input consists of translating uniform noise patterns, the speeds of which range from -6 to 6 pixels/frame, corresponding to phase gradients of $\pm 180°$ for the given filter pair. For noninteger speeds, we round off the displacement with respect to the first frame toward the nearest integer. For each of the 26 speeds, we have run 100 simulations with different random noise patterns. Fig. 1 shows the absolute value of the error on the phase gradient ("phase error;" median values with upper and lower quartiles). A breakdown in performance is observed for speeds corresponding beyond the $[-150°, 150°]$ range which is due to the phase wrap-around (similar breakdowns are observed for quadrature pairs that are tuned to different frequencies). Indeed, there is a region near $\phi_x = \pm 180°$, where the phase gradient does not yield stable results: small changes in phase can correspond to large spatial displacements due to the phase wrap-around. Furthermore, our simple phase-unwrapping technique is not suited for large phase gradients. In the second stage of our algorithm, these cases will be detected by examining the quality of the least-squares regression (see further). Since there is a breakdown in performance for phase gradients greater than a certain value, the filter frequencies impose an upper bound on the component velocity that can be reliably detected.

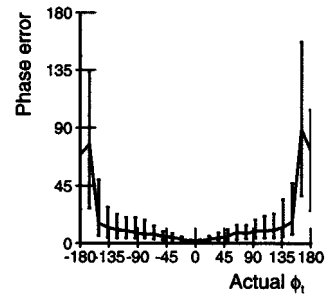

Fig. 1. Phase error, i.e., the absolute value of the error made on the phase gradient (median values with upper and lower quartiles, taken over a batch of 100 simulations per phase gradient) for the case of translating (x, t) uniform noise.

B. Confidence Measure

Barron *et al.* [1], [3] have found that the use of confidence measures, i.e., a measure for determining the correctness of the computed velocities, greatly influences the performances of different optical flow algorithms. In their implementations, they have used confidence measures as thresholds to retain a subset of *valid* estimates, although they note that in many applications all velocity estimates can be retained along with their respective confidence values which could, e.g., be used as weights in subsequent computations.

In their phase-based approach, Fleet and Jepson [5] originally proposed a two-fold constraint on the component velocity estimates, namely one on the local frequency, to ensure that the detected local frequency is within the passband of the spatiotemporal filter pair, and one on the local signal amplitude, to reject filter pair outputs where no significant power is present in that frequency region. In subsequent publications [6], [7], [15], they have extended the first constraint by detecting the "singularity neighborhoods," i.e., the regions where phase information becomes unstable. We have performed a similar simulation to those described in [6] and [7], [15] to illustrate this concept and to visualize the singularity neighborhoods in a Gabor scale-space framework. Consider a 1-D pattern of uniform noise, which is convolved with quadrature Gabor

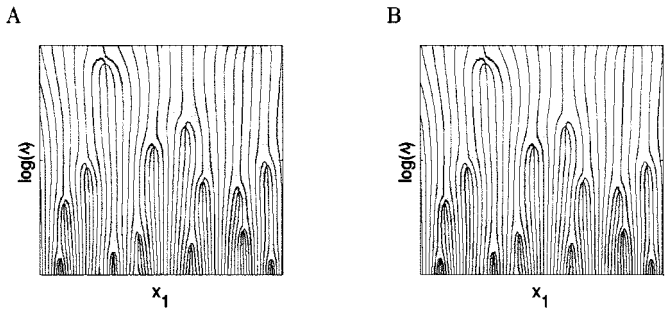

Fig. 2. Contour plot of the phase component of the Gabor scale space expansion of a uniform noise pattern. The horizontal and vertical axes represent spatial position x_1 and log of the scale λ (λ spans two octaves). The gray areas denote (a) the phase instability regions detected by Fleet and Jepson and (b) the regions detected using our nonlinearity criterion.

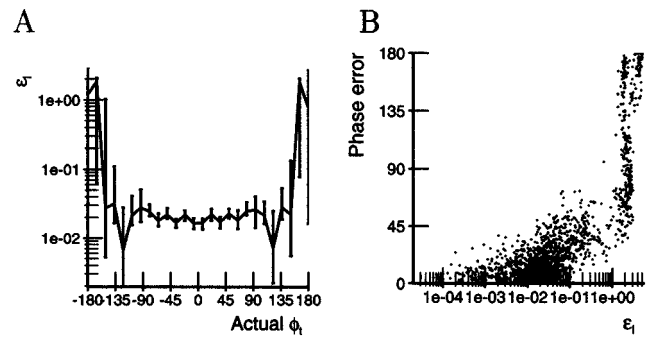

Fig. 3. (a) Phase nonlinearity measure, ε_l, as a function of the actual phase gradient for the case of translating (x, t) uniform noise (same as Fig. 1). The median values and upper and lower quartiles are plotted. (b) Scatter plot of ε_l and the phase error.

filter pairs of different scales, λ, inversely related to the center frequencies, with constant bandwidths of $\beta = 0.6$ octaves. The phase component of the quadrature pair is plotted in a contour plot where the horizontal and vertical axes denote the spatial position x_1 and the scale λ, the latter on a logarithmic axis (150 scales between 0.05 and 0.2). This allows for the visualization of the phase behavior with respect to scale and spatial position, corresponding to image expansion (looming) and translation in the optical flow case. Phase information yields reliable information for image matching (is "stable") if the equiphase contours are near-vertical in the Gabor scale-space expansion [15]. Fig. 2(a) shows that the phase structure is generally stable, except at several isolated regions where the equiphase contours become horizontal "singularities." Fleet and Jepson have proposed a straightforward heuristic to detect these "singularity neighborhoods." Fig. 2(a) shows the regions that are detected using this method (shaded gray). By detecting whether or not the filter pair output is located in a phase singularity neighborhood, the unreliable component velocities can be discarded.

In our approach, we estimate the phase *nonlinearity*, which is a major source of erroneous velocity estimates. There is a relationship between phase nonlinearity and phase instability. Indeed, one expects that, if the phase information becomes unreliable (unstable), it is not likely to have a linear evolution over time. Since we are considering spatial filter pairs and not spatiotemporal ones, we can easily measure the degree to which the phase is linear over time in the following way. We compute the phase for a given filter pair at all frames t that are considered for the optical flow estimation, and perform a linear least-squares regression on the (ϕ, t)-pairs (using the phase-unwrapping technique explained in Section II-A). The mean-square-error (MSE) divided by the absolute value of the estimated gradient (in radians), ε_l, yields a measure for the phase nonlinearity over time. In the Gabor scale space expansion, we compute this nonlinearity measure, ε_l, as a function of spatial position over a span of five pixels (note that, in the actual algorithm, it will be computed over time). Spatial positions where ε_l exceeds 10^{-5} are shaded gray in Fig. 2(b). It is clear that these areas roughly correspond to the phase instability regions in Fig. 2(a), although the exact shapes and sizes differ. In the subsequent simulations, the nonlinearity threshold value is larger, since subpixel displacements, as well as deviations from pure translational motion, can occur, leading to larger regression errors. It is important to note that this confidence measure also detects cases where the phase information is in reality linear, but where our phase-unwrapping technique yields incorrect results.

Fig. 3(a) shows the nonlinearity measure ε_l as a function of the phase gradient (median values and quartiles) for the same simulations shown in Fig. 1 (the (x, t)-case described in Section II-A). The lower ε_l-values for phase gradient of $\pm 122°$ stems from the fact that these phase gradients correspond to nearly integer displacements (4.08 pixels/frame). The breakdown in performance starts for phase gradients beyond the $[-150°, 150°]$ range (see Fig. 1). Beyond this range, as is evident from Fig. 3(a), the linear regression of phase over time yields a significantly higher ε_l (this has also been observed in simulations with filters tuned to different frequencies). This is further illustrated in Fig. 3(b), where the phase error is plotted as a function of ε_l. The majority of large phase errors correspond to values of ε_l greater than 0.01. We conclude that ε_l is a good indicator for the accuracy of the phase gradient estimate. Thus, we will reject component velocities for which the phase information is not linear over time, i.e., when ε_l exceeds a certain threshold, τ_l. The effect of the nonlinearity threshold, τ_l, on the accuracy and the density of the estimated optical flow field will be examined in Section III.

C. Full Velocity

One component velocity $\mathbf{v}_{c,i}$ only estimates the velocity component in the direction, orthogonal to the corresponding filter pair's orientation, i.e., in the direction of its spatial gradient. The 2-D (full) velocity can be determined if several such estimates are available. Thus, we use a bank of spatial quadrature filter pairs. Each filter pair yields a component velocity. Our filterbank consists of 11 quadrature Gabor filter pairs with bandwidths of $\beta = 0.6$ octaves and is depicted in Fig. 4 (frequency domain).

Every component velocity $\mathbf{v}_{c,i}$ constrains the full velocity $\mathbf{v}$ to lie on a "constraint line" L_i in a hypothetical velocity space, with an orientation orthogonal to $\mathbf{v}_{c,i}$ [4]. This line is defined as

$$L_i\colon \frac{\mathbf{v} \cdot \mathbf{v}_{c,i}}{\|\mathbf{v}_{c,i}\|} = \|\mathbf{v}_{c,i}\|.$$

If several component velocities with different orientations are present, the intersection of the corresponding constraint lines will yield the full velocity $\mathbf{v}$ called intersection-of-constraints

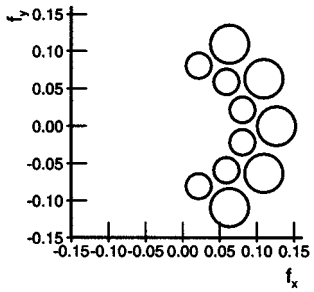

Fig. 4. Composition of our spatial filterbank in the frequency domain. The circles correspond to the 1 sigma-borders of the enveloping Gaussians in the frequency domain.

(IOC).[1] Since every component velocity forces the full velocity $\mathbf{v}$ to lie on a constraint line L_i, an overdetermined set of constraints is produced by the bank of quadrature pairs. Since the component velocity estimates are noisy, these lines do not intersect in a single point. Schunck [21] introduced his constraint line clustering procedure which determined the center of the 1-D cluster formed by the intersections of the constraint line, at a given point, with those corresponding to the local estimates around that point. Jepson and Black [16] went on and applied a statistical expectation–maximazation (EM) approach to a 2-D cluster detection process, which allowed them to determine the velocities of multiple objects. We will proceed differently and regard the constraint lines as soft constraints in an optimization process that determines $\mathbf{v}$. The optimization process is cast into a goal programming[2] format as follows. 1) The two vector components of the full velocity $\mathbf{v}$, namely u and v, correspond to the two variables in the goal programming problem. 2) Each constraint line corresponds to a separate goal, namely the minimization of the orthogonal distance between the constraint line and $\mathbf{v}$. The solution is found by striving toward all goals simultaneously, which can be achieved by minimizing F, the sum of all orthogonal distances.

Goal programming can be performed by the recurrent neural network suggested by Van Hulle [23], further called the goal programming network (GPN). It consists of two sets of amplifiers, the f- and g-amplifiers, the outputs of which represent the degree to which a goal is satisfied, and the variables of the goal programming problem, respectively. For our purpose, we configure the GPN in such a way that it converges to a state $\mathbf{v} = (u, v)$, where the summed orthogonal distance between $\mathbf{v}$ and the constraint lines is minimal. Contrary to the original GPN, we allow the variables u and v to become negative by replacing the transfer functions of the g-amplifiers by $g(x) = x$. Furthermore, the transfer functions of the f-amplifiers are replaced by $f(x) = x$. We update the network state at iteration step $k + 1$ as follows:

$$\mathbf{v}^{k+1} = \mathbf{v}^k - \Delta s \sum_{i=1}^{N} \mathbf{v}_{c,i} \left(\frac{\mathbf{v}^k \cdot \mathbf{v}_{c,i}}{\|\mathbf{v}_{c,i}\|} - \|\mathbf{v}_{c,i}\| \right)$$

[1]Note that the IOC principle is normally aimed at determining the *true* velocity from local velocities pooled from a spatial neighborhood, whereas we pool estimates from different filters at a single spatial location.

[2]Goal programming is similar to linear programming, but with the single objective replaced by several objectives or goals toward one must simultaneously strive [13].

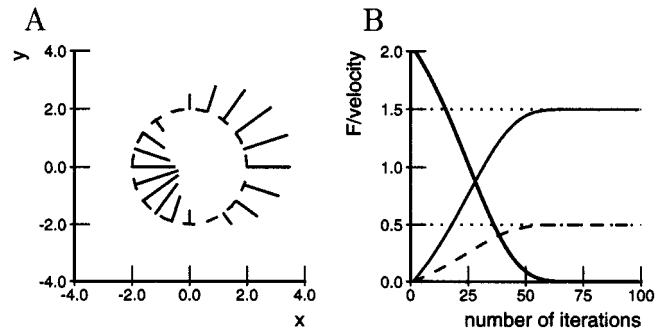

Fig. 5. (a) Illustration of the aperture problem. The circle (dashed line) is translating with a velocity $\mathbf{v} = [1.5; 0.5]$ pixels/frame. The observed velocities, when viewing through a small aperture at different spatial locations, are denoted by the oriented lines. (b) Evolution of the summed orthogonal distances (F; thick line, scaled by a factor of 0.1) and the states of the network u (thin, solid line) and v (thin, dashed line).

with Δs the time interval between two state updates in the GPN, and N the number of constraint lines. In order to increase the speed of convergence, we make Δs adaptive using the following simple heuristic: if F decreases, we increase the time step, $\Delta s^{k+1} = 1.05 \cdot \Delta s^k$, otherwise we undo the last state update and set $\Delta s^{k+1} = 0.5 \cdot \Delta s^k$.

As a demonstration to the intersection-of-constraints principle using a GPN, we describe a solution to the spatial aperture problem. Consider the circle shown in Fig. 5(a), translating with a velocity $\mathbf{v} = [1.5; 0.5]$ pixels/frame. When viewed through small apertures at different positions, the partial contours seemingly move orthogonal to their orientations, as indicated by the vector lines in Fig. 5(a) for 16 points on the circle. Every velocity vector constrains the true velocity $\mathbf{v}$ and these constraints are hard-wired into the GPN. We use an initial time step $\Delta s = 0.001$ and set the initial state of the network to [0; 0]. The algorithm ends when the standard deviation of F, computed over the last ten iterations, does not exceed 10^{-10}. In this simulation, the GPN finds the correct velocity vector ($\mathrm{MSE} = 4.93\,10^{-32}$) after 98 iterations. Fig. 5(b) shows the evolution of the summed orthogonal distances (scaled by a factor of 0.1), F (thick line, starting at 2.0 and decreasing), and the network states, corresponding to the horizontal (thin, solid line) and vertical component (thin, dashed line) of the velocity, both starting at zero and increasing.

We test this strategy for estimating the velocities of translating 2-D uniform noise patterns, using the filterbank shown in Fig. 4. The noise patterns translate in 16 different directions at 20 different speeds (logarithmically spaced between 0.2 and 5 pixels/frame), and we run 50 simulations with different random noise patterns for every combination. In this simulation, we do not use a threshold to reject unreliable component velocities. Performance is measured in terms of the error measure proposed by Fleet and Jepson [5], namely the angular error between the space–time direction vectors $(\mathbf{v}_{\mathrm{act}}, 1)$ and $(\mathbf{v}, 1)$

$$\psi_e = \arccos \left[\frac{(\mathbf{v}_{\mathrm{act}}, 1) \cdot (\mathbf{v}, 1)}{\sqrt{1 + \|\mathbf{v}_{\mathrm{act}}\|^2} \sqrt{1 + \|\mathbf{v}\|^2}} \right] \tag{3}$$

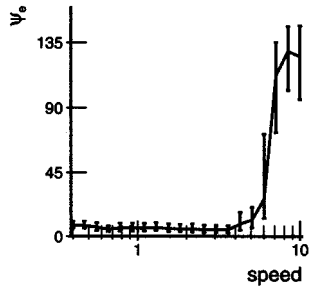

Fig. 6. Results (median values and upper and lower quartiles) for the translating uniform noise patterns as a function of speed.

where $\mathbf{v}_{\text{act}}$ is the actual 2-D velocity and $\mathbf{v}$ is the full velocity estimate.[3] Fig. 6 summarizes the results. There is a breakdown in performance for speeds greater than 3.5 pixels/frame and there is no systematic difference in performance between different directions (results not shown). The breakdown corresponds to what we expect: for the second scale of filters, with a wavelength of 7.92 pixels, this corresponds to a phase gradient of 159° (comparable to the 150° of the (x, t)-example in Section II-A). Note that a part of the error is introduced by the quantization in the time domain, due to which the noise pattern makes discrete jumps over time.

For reasons of efficiency, the filter outputs are computed using cascaded 1-D convolutions, rather than computationally expensive 2-D convolutions, as suggested by Heeger [12]. The length of a 1-D template is is set to 6 standard deviations of the Gaussian envelope of the corresponding filter pair. The filter pair outputs at one time frame are computed by zero-padding the image and performing the 1-D convolutions. This can result in erroneous velocity estimates at the image borders, due to the discontinuity introduced by zero-padding. Therefore, we discard a border region demarcated by the point at which the spatial Gaussian envelope reaches 10% of its peak level, i.e., at $\sigma_i \sqrt{\log_e(100)}$ pixels.

III. Results

We test our algorithm on the image sequences that have been used by Barron *et al.* [1], [3] and Galvin *et al.* [9], [19] for comparing performances between different optical flow algorithms. We include some of these performances in order to better interpret our results. The performances are given as the mean and standard deviation of the angular error ψ_e (3), as well as the densities of the resulting flow field estimates (the border region is excluded to compute the density). The image sequences are divided into two groups: the synthetic image sequences, for which the 2-D optical flow fields are known, and the real image sequences. For the first group of sequences, we use a very tight stability criterion ($\tau_l = 0.005$) and only compute 2-D velocities if there are at least $N_{\min} = 7$ valid component velocities available. For the second group, we relax these constraints to $\tau_l = 0.05$ and a minimum of $N_{\min} = 5$ valid component velocities. For all simulations, we estimate the temporal phase gradients over five frames. Unless stated, we refer to [1] and [3] for simulation details for the other algorithms. The optical flow fields, both actual and estimated, are scaled and subsampled for visualization purposes. Flow fields for the same image sequence are scaled and subsampled using the same factors.

[3]We will assume that the output of the GPN is an estimate of the actual 2-D velocity. Thus, we neglect the spatial aperture problem. This is a reasonable assumption, since the spatial extent of our filters is fairly large (see Section V).

TABLE I
Results for the Translating Square Sequence

Technique	Av. Err.	St. Dev.	Dens.
Horn & Schunck (modified) $\|\nabla I\| \geq 1.0$	26.46°	10.86°	42.9%
Lucas & Kanade ($\lambda_2 \geq 1.0$)	0.21°	0.16°	7.9%
Uras *et al.* ($\det(H) > 1.0$)	0.15°	0.10°	26.1%
Nagel $\|\nabla I\|_2 \geq 1.0$	26.67°	11.87°	44.0%
Anandan (unthresholded)	31.46°	18.31°	100%
Singh (Step 2, $n = 2$, $w = 2$)	45.16°	21.10°	100%
Heeger	6.16°	4.02°	29.3%
Fleet & Jepson $\tau = 2.5$	0.18°	0.13°	12.6%
Gautama & Van Hulle	5.21°	2.12°	68.7%

A. Synthetic Image Sequences

1) Translating Square Sequence: This sequence consists of a translating dark square, 40 pixels wide, on a white background ($v = [1.33, 1.33]$). It is constructed by downsampling a larger version which moves at an integer velocity. It has been used by Barron *et al.* to demonstrate the spatial aperture problem.

Due to the large spatial extent of our filters, our approach does not suffer from the aperture problem and yields 2-D velocity, rather than normal velocity estimates. However, if the square were larger, our approach would yield normal velocities at square's sides, resulting in large errors. Indeed, the low performances of some techniques are due to the inability to discriminate between 2-D velocity and normal components. Table I summarizes our results and those obtained by Barron *et al.* [1], [3].

2) Translating and Diverging Tree Sequences: The Translating Tree sequence simulates a translational camera movement along the y-axis, orthogonal to the line of sight, while capturing a textured, planar surface (the picture of a tree), which is slanted with respect to the fronto-parallel plane. The velocities are parallel to the image x-axis and range from 1.73 to 2.26 pixels/frame. In the Diverging Tree sequence, the same planar surface is captured by a camera that is translating along the line of sight, resulting in a looming effect, with the center of expansion in the middle of the image. The speeds vary from 1.29 pixels/frame on the left side, to 1.86 pixels/frame on the right side of the image.

The error statistics for the different methods are included in Table II. The errors for these sequences are rather large, compared to some of the results by Barron *et al.* [1], [3]. Since these image sequences have smooth flow fields, this is possibly due to

TABLE II
RESULTS FOR THE TRANSLATING AND DIVERGING TREE SEQUENCE

Technique	Translating Tree			Diverging Tree		
	Av. Err.	St. Dev.	Dens.	Av. Err.	St. Dev.	Dens.
Horn & Schunck (modified) $\|\nabla I\| \geq 5.0$	1.89°	2.40°	53.2%	2.50°	3.89°	32.9%
Lucas & Kanade ($\lambda_2 \geq 1.0$)	0.66°	0.67°	39.8%	1.94°	2.06°	48.2%
Uras *et al.* ($\det(H) > 1.0$)	0.46°	0.35°	41.8%	3.83°	2.19°	60.2%
Nagel $\|\nabla I\|_2 \geq 5.0$	2.24°	3.31°	53.2%	2.94°	3.23°	100%
Anandan	4.54°	3.10°	100%	7.64°	4.96°	100%
Singh (Step 2, $n = 2$, $w = 2$, $\lambda_1 \leq 0.1$)	1.11°	0.89°	99.6%	8.40°	4.78°	99.0%
Heeger	4.53°	2.41°	57.8%	4.95°	3.09°	73.8%
Fleet & Jepson $\tau = 2.5$	0.32°	0.38°	74.5%	0.99°	0.78°	61.0%
Gautama & Van Hulle	2.67°	1.43°	67.8%	4.07°	2.42°	77.3%

the spatial integration of local estimates by the other techniques (either gradients or component velocities at different spatial locations), which is absent in our approach (we only pool component velocities from a single spatial location).

3) Yosemite Fly-Through Sequence: The Yosemite fly-through sequence is the more complex test case of the synthetic image sequences. Each frame has been generated by mapping an aerial photograph onto a digital-terrain map [12]. Speeds in the lower left corner go up to four pixels/frame and the clouds translate to the right at one pixel/frame while changing shape, due to which their true motion is not simply related to image brightness changes. The middle frame of the image sequence is shown in Fig. 7(a), and the actual flow field in Fig. 7(b).

The result shown in Fig. 7(d) illustrates the efficacy of our phase nonlinearity criterio. Since the clouds are changing shape over time, the phase information is not linear and, as a consequence, results in unreliable estimates. Most of the velocities above the mountain rim have indeed been rejected by our nonlinearity criterion. Also, the speeds in the lower left are too high to be captured by our filters: the breakdown in performance for our filterbank lies around 3.5 pixels/frame (see Section II-C), beyond which point, the estimated phase is no longer linear over time (at least not given our simple phase-unwrapping technique). [For this sequence we relax our nonlinearity criterion slightly to $\tau_l \leq 0.01$: otherwise, the density of the resulting flow is only 17.8%, with an angular error of $3.50° \pm 2.49°$; see Fig. 7(c).] The error statistics are shown in Table III.

Other algorithms have been described with much lower errors and higher densities than those in Table III, such as the "Skin and Bones" algorithm by Ju *et al.* [17] and the "Total Least-Squares-Based Optic Flow" technique developed by Bab-Hadiashar and Suter [2]. The first yields an angular error of $2.16° \pm 2.0°$ with a density of 100%, while the second yields an error of $1.97° \pm 1.96°$ with a density of 72.0%. However, in both cases, the clouds have been masked, and the performances for the original image sequence, i.e., with the cloud region, have not been reported. If the clouds are masked, our algorithm yields and error of $3.46° \pm 2.15°$ (this is not significantly better than for the original image sequence, since our nonlinearity criterion rejects most velocity estimates in the cloud region).

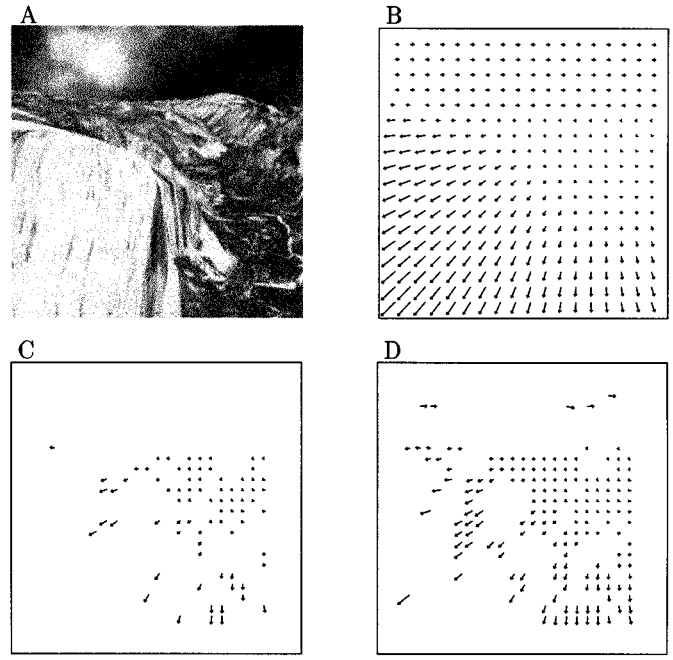

Fig. 7. (a) Middle frame of the Yosemite fly-through sequence. (b) Actual flow field. (c) and (d) Estimated flow fields with $\tau_l = 0.005$ and $\tau_l = 0.01$.

TABLE III
RESULTS FOR THE YOSEMITE SEQUENCE

Technique	Av. Err.	St. Dev.	Dens.
Horn & Schunck (modified) $\|\nabla I\| \geq 5.0$	5.48°	10.41°	32.9%
Lucas & Kanade ($\lambda_2 \geq 1.0$)	4.10°	9.58°	35.1%
Uras *et al.* ($\det(H) > 1.0$)	6.73°	16.01°	14.7%
Nagel $\|\nabla I\|_2 \geq 5.0$	6.03°	11.04°	32.9%
Anandan	15.84°	13.46°	100%
Singh (Step 2, $n = 2$, $w = 2$, $\lambda_1 \leq 0.1$)	12.90°	11.57°	97.8%
Heeger (combined)	11.74°	19.04°	44.8%
Fleet & Jepson $\tau = 2.5$	4.29°	11.24°	34.1%
Gautama & Van Hulle	4.40°	3.74°	34.7%

4) Complex Synthetic Image Sequences: We test our technique further on two complex, synthetic image sequences developed by Galvin [9], [19], namely the Street sequence and the Office sequence, for which they provide a ground truth flow field. For every sequence, 20 frames are available via public ftp[4] (we have used frames 98–102). We have also estimated the flow fields using Fleet and Jepson's technique (using frames 90–110) with a spatiotemporal Gaussian smoothing kernel with $\sigma = 2.5$ pixels (frames), and τ set to 1.25 and 2.5, and Lucas and Kanade's (using frames 93–107), which is the best performing technique in the studies by Galvin *et al.* The results are shown in Table IV. The Street sequence has also been used for testing Weickert and Schnörr's algorithm (2001), which is a differential technique that imposes a spatiotemporal smoothness constraint

[4]http://www.cs.otago.ac.nz/research/vision/Downloads/.

TABLE IV
RESULTS FOR THE STREET AND OFFICE SEQUENCES

Technique	Street			Office		
	Av. Err.	St. Dev.	Dens.	Av. Err.	St. Dev.	Dens.
Lucas & Kanade ($\lambda_2 \geq 1.0$)	4.83°	5.42°	30.5%	7.45°	6.89°	15.5%
Fleet & Jepson ($\tau = 1.25$)	3.12°	3.13°	20.5%	10.33°	19.32°	5.4%
Fleet & Jepson ($\tau = 2.5$)	3.15°	4.40°	23.2%	6.81°	12.07°	8.6%
Gautama & Van Hulle	4.64°	2.48°	23.7%	9.59°	6.34°	21.4%

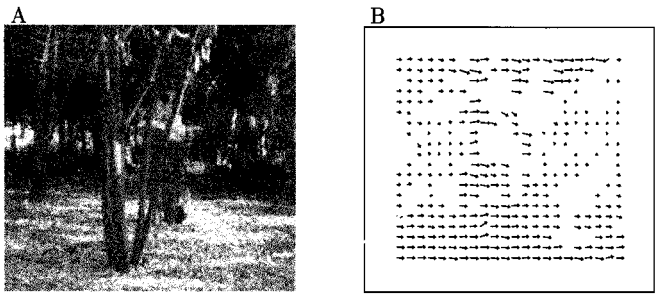

Fig. 8. (a) Middle frame and (b) estimated flow field for the SRI sequence. The estimated flow field has a density of 71.3%.

on the flow field. They report an average angular error of 4.85°, with a density of 100%.

B. Real Image Sequences

Since there are no actual flow fields available for these image sequences, it is impossible to quantitatively evaluate our results. For every sequence, we will show the middle frame, give the short description given by Barron *et al.* [1], [3], and compare this to our estimated flow field. The first two sequences are examples of global motion (resulting from camera motion), whereas the other two result from moving objects. For a detailed description of the results with other techniques, see [1] and [3].

1) SRI Sequence: The camera translates parallel to the ground plane, perpendicular to the line of sight, in front of clusters of trees. It is a challenging sequence because of the relatively poor resolution, the amount of occlusion, and the low contrast. Velocities are as large as two pixels/frame. See Fig 8.

The estimated velocities in the slanted ground plane correspond to what is expected, i.e., a linear gradient. The velocities in the background are a lot smaller than those of the foremost tree cluster. The latter velocities are orthogonal to the branches' orientations, due to the spatial aperture problem (which is also the case in other techniques described by Barron *et al.*)

2) NASA Sequence: This sequence is primarily dilational. The camera moves along its line of sight toward the Coca Cola can near the center of the image. Image velocities are typically less than one pixel/frame.

The resulting flow field is indeed dilational [Fig. 9(b)]. Compared to other techniques, such as those by Lucas *et al.*, our technique yields a very dense flow field, namely one of 87.9%, compared to 35.3% and 13.3%, respectively. The speeds are below one pixel/frame.

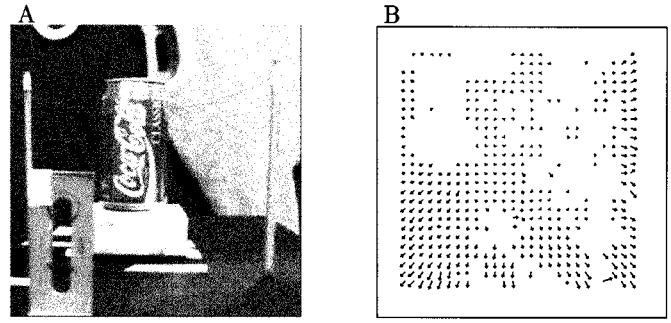

Fig. 9. (a) Middle frame and (b) estimated flow field for the NASA sequence. The estimated flow field has a density of 87.9%.

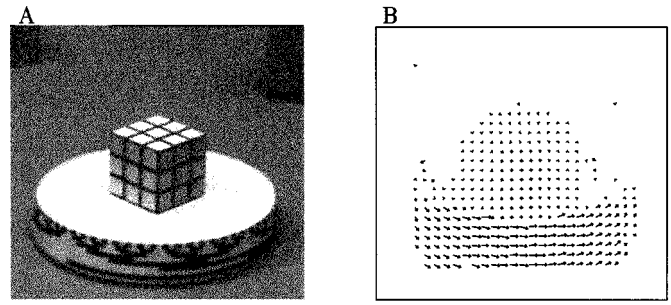

Fig. 10. (a) Middle frame and (b) estimated flow field for the Rotating Rubik's Cube sequence. The estimated flow field has a density of 70.0%.

3) Rotating Rubik's Cube Sequence: This sequence shows a Rubik's cube which is rotating counter-clockwise on a turntable. The motion field induced by the rotation of the cube includes velocities less than two pixels/frame. Velocities on the turntable range from 1.2 to 1.4 pixels/frame, and those on the cube are between 0.2 and 0.5 pixels/frame.

Fig. 10(b) shows that almost all of the valid velocity estimates are located on the side of the turntable (due to the rotation), on the turntable itself and on the cube. Indeed, besides these objects, there are no moving objects. The velocity estimates on the cube roughly range from 0.1 to 0.5 pixels/frame. The velocities go up to 1.5 pixels/frame.

4) Hamburg Taxi Sequence: In this street scene, there are four moving objects: 1) the taxi turning the corner; 2) a car in the lower left, driving from left to right; 3) a van in the lower right driving right to left; and 4) a pedestrian in the upper left. Image speeds of the four moving objects are approximately 1.0, 3.0, 3.0, and 0.3 pixels/frame, respectively.

Fig. 11 shows that all four objects are successfully detected by our technique. We have manually segmented the velocity estimates to the corresponding objects on the basis of their direction θ, speed and spatial location [see grayscale coding of arrows in Fig. 11(b)]. The mean and standard deviations of their speeds and directions are the following: 1) 0.71 ± 0.22 pixels/frame with $\theta = 141.32° \pm 17.38°$ for the taxi turning the corner (3929 pixels); 2) 2.56 ± 0.29 pixels/frame with $\theta = -9.59° \pm 5.036°$ for the car in the lower left (1726 pixels); 3) 2.49 ± 0.23 pixels/frame with $\theta = 179.32° \pm 7.41°$ for the van in the lower right (642 pixels); and 4) 0.24 ± 0.04 pixels/frame with $\theta = 187.96° \pm 18.86°$ for the pedestrian (373 pixels).

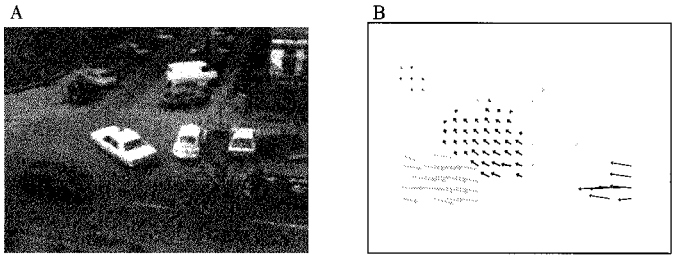

Fig. 11. (a) Middle frame and (b) estimated flow field for the Hamburg Taxi sequence. The estimated flow field has a density of 24.5%. The arrow colors in (b) correspond to the manual label each velocity has received (there are four objects in total), based on position, direction, and speed.

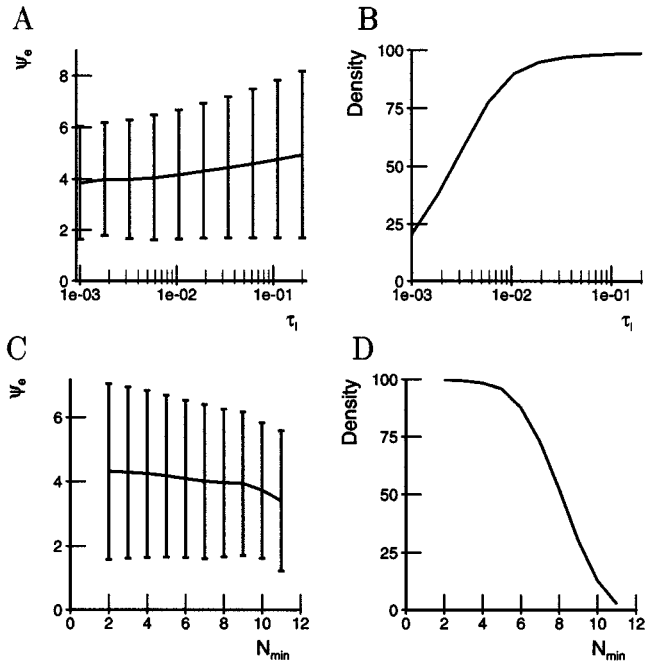

Fig. 12. Sensitivity analysis on the Diverging Tree sequence. (a) Evaluation of the performance (mean angular error and standard deviation) as a function of the nonlinearity measure ε_l. (b) Evaluation of the performance (density) as a function of the nonlinearity measure ε_l. (c) and (d) Evaluation of the performance as a function of the minimal number of valid component estimates $N_{\min}$.

IV. Sensitivity Analysis

In order to examine the influence of the parameters on the performance of our technique, we plot the error performance and density of the estimated flow field for the Diverging Tree sequence as a function of the phase nonlinearity threshold, τ_l [Fig. 12(a) and (b)], and as a function of the minimal number of valid component velocities, $N_{\min}$ [Fig. 12(c) and (d)]. Fig. 12(a) and (b) show the mean angular errors and standard deviations for different values of the phase nonlinearity threshold, τ_l, with $N_{\min} = 7$. As the nonlinearity criterion is relaxed, both the mean error and standard deviation increase slightly (from $3.83° \pm 2.21°$ to $4.93° \pm 3.25°$), whereas the density increases considerably. Similarly, as is shown in Fig. 12(c) and (d), when the minimal number of valid component velocities is increased (keeping $\tau_l = 0.005$), the mean and standard deviation of the angular error decrease slightly (from $4.31° \pm 2.74°$ to $3.40° \pm 2.18°$) and the density is greatly influenced. Both simulations illustrate the tradeoff between accuracy and density, the preference for which is application-specific. In addition, the parameter values that achieve a certain accuracy/density are image sequence specific as well.

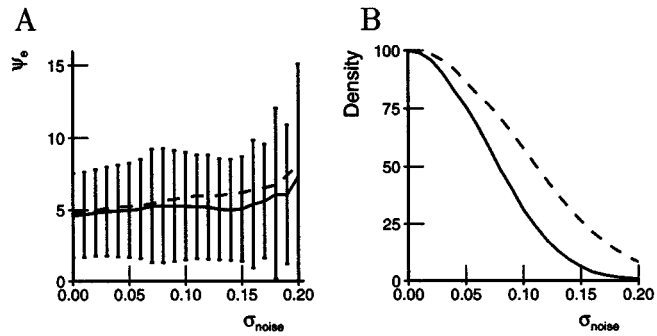

Fig. 13. (a) Error performance (mean angular error and standard deviation) and (b) density of the estimated flow field for the Diverging Tree sequence, as a function of the noise level, σ_{noise} (expressed as a fraction of the maximum gray level in the image sequence. In both figures, two levels of τ_l have been used: $\tau_l = 0.05$ (solid line) and $\tau_l = 0.1$ (dashed line).

We have further evaluated the performance of our technique in the presence of additive Gaussian noise, with zero mean and standard deviation σ_{noise}, expressed as a fraction of the maximum amplitude of the stimulus. We have evaluated the angular error using $\tau_l = 0.05$ and $N_{\min} = 5$. The error performance remains fairly constant with increasing levels of noise [Fig. 13(a), solid line and error bars] and, again, the density is greatly influenced by the presence of noise [a density of 50% is reached for $\sigma_{\text{noise}} = 0.08$; Fig. 13(b), solid line]. If the nonlinearity threshold is relaxed to $\tau_l = 0.1$, the error performance deteriorates slightly [Fig. 13(a), dashed line, the error bars are not included for clarity's sake], but the density increases significantly. This indicates that, although the separate phase estimates at a given time frame are subject to noise, the slope of the regression line remains accurate. Thus, the problem is that the increase of ε_l can be due to the presence of noise or due to the nonlinearity of the phase information at that spatial location.

V. Discussion

From a computational point of view, our approach is comparable to Fleet and Jepson's [5], albeit that we perform a spatial convolution at every time frame, rather than a single spatiotemporal convolution. For the sake of comparison, we give the computation times for the Hamburg Taxi and the NASA sequences for both techniques. One should, however, take into account that our technique has been implemented in Matlab, whereas we have used Fleet and Jepson's C-implementation, which is publicly available. Furthermore, we only use five time-frames for the computation, whereas Fleet and Jepson use 21 frames. It is also important to note that the computation times for both implementations are highly dependent on the densities of the resulting flow fields, since only for those points, the full velocity is determined. Our technique estimates the flow field for the Hamburg Taxi sequence in 68 s (density of 21.1%) and the NASA sequence in 205 s (density of 60.7%), while Fleet and Jepson's technique needs 104 (density of 27.0%) and 202 s (density of 13.3%). As mentioned in their paper, the computational

load can be significantly reduced (also for our technique) by computing the component velocities on a discrete spatial grid, e.g., by computing a complex filter output every σ_i pixels, and, furthermore, a number of 1-D convolution results can be used multiple times. Our approach offers the advantage that the computations are purely local (including the stability criterion) and that it can, therefore, be implemented very efficiently if only a subsampled version of the optical flow field is required.

Another advantage of our technique is that it allows for arbitrary temporal spans over which the flow field is computed. In the simulations that have been performed, we use a span of five time frames. To examine the effect it has on the performance, we increase this number for the Diverging Tree sequence (adjusting τ_l such the density remains around 75%). Initially, there is a slight decrease in average angular error of 0.3°, after which the performance remains constant. A longer temporal span might even cause problems due to our nonlinearity measure: if the translating object (or texture) moves beyond a filter pair's spatial extent within the temporal span, the phase information of that filter pair becomes nonlinear and will be rejected. Indeed, for the Diverging Tree sequence, we need to relax the nonlinearity criterion in order to obtain the same density for longer time spans. In our opinion, this is true for other techniques as well: the lower densities for Fleet and Jepson's and Lucas and Kanade's techniques could be due to the long time spans over which the flow field is estimated (21 and 15 frames, respectively).

Since our technique uses an approach which is related to Fleet and Jepson's, namely a phase-based one, we highlight some of the differences. Fleet and Jepson use spatiotemporal filter pairs to estimate the phase gradient and use a stability criterion that, among others, ensures that the instantaneous frequency [8] lies within the passband of the filter pairs. This means that they have to *tile* the spatiotemporal frequency space with their filters, and that their filter pairs are *velocity*-tuned. Our approach spatially filters the images, due to which only the spatial frequency domain needs to be considered, and due to which our spatial filter pairs are able to estimate the phase gradients irrespective of the velocity at that spatial location. Since Fleet and Jepson use a spatiotemporal radially symmetric Gabor filter, the temporal span over which the optical flow field is estimated, is a function of the choice of their filterbank, which is not the case with our algorithm. Furthermore, in order to derive the full velocity, Fleet and Jepson's technique pools different component velocity estimates from a small spatial neighborhood, whereas our technique limits itself to estimates at a single location. As mentioned in the Results section, our approach is outperformed by far for the Translating and Diverging Tree sequences, possibly due to this spatial pooling. The latter strategy could easily be incorporated in our technique. However, in that case we would forfeit the purely local aspect of the computations, which are an asset for an efficient parallel implementation and the fast computation of subsampled optical flow fields. Also, our stability criterion examines the linearity of the phase over time (local computation), whereas Fleet and Jepson detect neighborhoods around phase singularities, requiring spatial derivatives, which cannot be computed purely locally.

Finally, our technique assumes the optical flow field to be uniform within the spatial extent of our filter pairs (*translational* model; for an overview, see [22]). A more sophisticated scheme, such as the *affine* model of motion, could be used, which would result in a quadratic, rather than a linear regression of the phase over time. The constraints corresponding to the resulting component *motion fields* would, however, become more complicated. All these are topics for further research.

VI. Conclusion

We have introduced a novel phase-based technique for estimating the optical flow field. It allows for a direct measurement of the phase nonlinearity, which is a good indicator for the reliability of the velocity estimate. At a single spatial location, several component velocities are estimated using quadrature Gabor filter pairs, each imposing a constraint on the full velocity at that point. A recurrent neural network determines the full velocity that is consistent with these constraints. Our approach has been tested on various image sequences, both synthetic and realistic, and the results have been compared to those of other techniques. However, it is difficult to rank different techniques, since their performances depend on different factors (the average and standard deviation of the error and the density of the estimated flow field), and since they can vary substantially for different image sequences. Overall, our technique ranks among the best ones we have tested, however, there is room for further improvement.

Acknowledgment

The authors wish to thank J. Barron *et al.* for making their source code and image sequences publicly available, and the Computer Graphics and Vision Research Group, University of Otago[5] for providing us with their image sequences.

References

[1] J. L. Barron, D. J. Fleet, S. Beauchemin, and T. Burkitt, "Performance of Optical Flow Techniques," Dept. Comput. Sci., Univ. Western Ontario, London, ON, Canada; and Dept. Comput. Sci., Queens Univ., Kingston, ON, Canada, Tech. Rep. TR 299, RPL-TR-9107, July 1992. Revised July 1993.

[2] A. Bab-Hadiashir and D. Suter, "Robust optic flow computation," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 59–77, 1998.

[3] J. L. Barron, D. J. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, 1994.

[4] C. L. Fennema and W. B. Thompson, "Velocity determination in scenes containing several moving images," *Comput. Graph. Image Processing*, vol. 9, pp. 301–315, 1979.

[5] D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *Int. J. Comput. Vis.*, vol. 5, no. 1, pp. 77–104, 1990.

[6] ——, "Phase-based disparity measurement," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 198–210, 1991.

[7] ——, "Stability of phase information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1253–1268, Dec. 1993.

[8] D. Gabor, "Theory of communication," *J. Inst. Elect. Eng.*, vol. 93, pp. 429–457, 1946.

[9] B. Galvin, B. McCane, K. Novins, and S. Mills, "Recovering motion fields: An evaluation of eight optical flow algorithms," in *Proc. British Machine Vision Conf.*, 1998, pp. 195–204.

[10] D. C. Ghiglia and L. A. Romero, "Robust two-dimensional weighted and unweighted phase unwrapping using fast transforms and iterative methods," *J. Opt. Soc. Amer. A*, vol. 11, pp. 107–117, 1994.

[5]http://www.cs.otago.ac.nz/gpxpriv/public_html/homepage.html.

[11] R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: Two-dimensional phase unwrapping," *Radio Sci.*, vol. 32, pp. 713–720, 1988.
[12] D. J. Heeger, "Model for the extraction of image flow," *J. Opt. Soc. Amer. A*, vol. 4, no. 8, pp. 1455–1471, 1987.
[13] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*. Oakland, CA: Holden-Day.
[14] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185–203, 1981.
[15] A. D. Jepson and D. J. Fleet, "Phase singularities in scale-space," *Image Vis. Comput.*, vol. 9, no. 5, pp. 338–343, 1991.
[16] A. Jepson and M. Black, "Mixture models for optical flow," *Proc. IEEE Comput. Vision Pattern Recognition Conf.*, pp. 760–761, 1993.
[17] S. X. Ju, M. J. Black, and A. D. Jepson, "Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency," in *Proc. IEEE Comput. Vision Pattern Recognition Conf.*, San Francisco, CA, 1996, pp. 307–314.
[18] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artificial Intell. (IJCAI)*, 1981, pp. 674–679.
[19] B. McCane, B. Galvin, and K. Novins, "On the evaluation of optical flow algorithms," in *Proc. 5th Int. Conf. Contr., Automat., Robot., Vision*, Singapore, 1998, pp. 1563–1567.
[20] C. Prati, M. Giani, and N. Leurattu, "SAR interferometry: A 2D phase unwrapping technique based on phase and absolute value informations," in *Proc. IGARSS'90*, 1990, pp. 2043–2046.
[21] B. G. Schunck, "Image flow segmentation and estimation by constraint line clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 1010–1027, Oct. 1989.
[22] C. Stiller and J. Konrad, "Estimating motion in image sequences, a tutorial on modeling and computation of 2D motion," *IEEE Signal Processing Mag.*, vol. 16, pp. 70–91, 1999.
[23] M. M. Van Hulle, "A goal programming network for linear programming," *Biol. Cybern.*, vol. 65, pp. 243–252, 1991.
[24] J. Weickert and C. Schnörr, "Variational optic flow computation with a spatio-temporal smoothness constraint," J. Math. Imaging Vis., 2001, to be published.

Temujin Gautama received the B.Sc. degree in electronic engineering from Groep T, Technologische Hogeschool, Leuven, Belgium, and the Master's degree in artificial intelligence from the Katholieke Universiteit, Leuven. He is currently pursuing the Ph.D degree at the Laboratorium voor Neuro- en Psychofysiologie, Medical School, Katholieke Universiteit.

His research interests include nonlinear signal processing, biological modeling, and self-organizing neural networks and their application to data mining.

Marc M. Van Hulle (M'97–SM'00) received the M.Sc. degree in electrotechnical engineering (electronics) and the Ph.D. degree in applied sciences from the Katholieke Universiteit, Leuven, Belgium, in 1985 and 1990, respectively. He also holds B.Sc.Econ. and MBA degrees from Limburgs Universitair Centrum, Belgium.

In 1992, he was with the Brain and Cognitive Sciences Department, Massachusetts Institute of Technology (MIT), Cambridge, as a Postdoctoral Scientist. He is affiliated with the Laboratorium voor Neuro- en Psychofysiologie, Medical School, Katholieke Universiteit, as an Associate Professor. He is also Founder and Director of Synes N.V., the data mining spinoff of the Katholieke Universiteit (http://www.synes.com). He has authored the monograph *Faithful Representations and Topographic Maps: From Distortion- to Information-Based Self-Organization* (New York: Wiley, 2000), and more than 60 technical publications. His research interests include neural networks, biological modeling, vision, data mining, and signal processing.

Dr. Van Hulle is an Executive Member of the IEEE Signal Processing Society, Neural Networks for Signal Processing (NNSP) Technical Committee (1996–1999, 2000–2003), the Publicity Chair of NNSP's 1999 and 2000 workshops, and the Program Co-Chair of NNSP's 2001 workshop, and reviewer and Coeditor of several special issues for several neural network and signal processing journals.