Fast two-frame multiscale dense optical flow estimation using discrete wavelet filters

Haiying Liu, Rama Chellappa, and Azriel Rosenfeld

Center for Automation Research, University of Maryland, College Park, Maryland 20770

Received October 10, 2002; revised manuscript received April 1, 2003; accepted April 2, 2003

A multiscale algorithm with complexity $O(N)$ (where N is the number of pixels in one image) using wavelet filters is proposed to estimate dense optical flow from two frames. Hierarchical image representation by wavelet decomposition is integrated with differential techniques in a new multiscale framework. It is shown that if a compactly supported wavelet basis with one vanishing moment is carefully selected, hierarchical image, first-order derivative, and corner representations can be obtained from the wavelet decomposition. On the basis of this result, three of the four components of the wavelet decomposition are employed to estimate dense optical flow with use of only two frames. This overcomes the "flattening-out" problem in traditional pyramid methods, which produce large errors when low-texture regions become flat at coarse levels as a result of blurring. A two-dimensional affine motion model is used to formulate the optical flow problem as a linear system, with all resolutions simultaneously (i.e., coarse-and-fine) rather than the traditional coarse-to-fine approach, which unavoidably propagates errors from the coarse level. This not only helps to improve the accuracy but also makes the hardware implementation of our algorithm simple. Experiments on different types of image sequences, together with quantitative and qualitative comparisons with several other optical flow methods, are given to demonstrate the effectiveness and the robustness of our algorithm.

OCIS codes: 100.2000, 100.2960, 150.4620.

1. INTRODUCTION

Research on optical flow, a concept originated by Gibson,[1] has progressed for more than 30 years. Accurate optical flow estimation is critical to many central applications in computer vision,[2] such as structure from motion,[3] motion detection,[4] locating the focus of expansion,[5] guiding an observer's heading direction,[6] segmenting independently moving objects,[7] extracting boundaries,[8] and analyzing medical video.[9] Psychophysical experiments on human subjects reveal that the first stage of processing in the human visual system is estimation of the motion field.[10,11]

Many methods of flow estimation have been proposed in the literature. Barron *et al.*[12] gave an empirical comparison of optical flow estimation techniques up to 1994. The methods were roughly divided into four categories:

1. Differential techniques. This approach is essentially based on the constant-intensity constraint. It computes the optical flow from spatiotemporal derivatives of the image intensity or after (low-pass, bandpass, etc.) filtering. Sobey and Srinivasan[13,14] extended it to a generalized gradient scheme using six different spatiotemporal filters in parallel derived from two linearly independent spatiotemporal kernels. The accuracy of the result depends heavily on accurate estimation of the derivatives. This requires the image intensity to be differentiable. In the case of two frames, it means highly oversampled images or nearly linear intensity structure, which is not always practical.

2. Region-based matching. This is regarded as a natural alternative to optical flow estimation[12] when the image sequence is not differentiable as a result of noise, small numbers of available frames, or undersampling (aliasing). The optical flow $\mathbf{u} = (u_x, u_y)$ is defined as a shift that maximizes a similarity measure between image regions at different times.

3. Energy-based methods. This approach computes optical flow in the frequency domain by using velocity-tuned filters. The optical flow is determined by fitting the spatiotemporal motion energy to a plane in frequency space. It can be shown that some of the energy-based methods are equivalent to correlation-based methods and to certain gradient-based methods.[12]

4. Phase-based techniques. This approach regards the optical flow as the phase change in spatiotemporal filter outputs. Although phase analysis is claimed to be more stable than its amplitude counterpart, it can also be unstable when phase singularities occur.

Optical flow techniques proposed since 1994 can be more or less classified into these four categories. To list a few: those of Refs. 15–18 are region-based matching, those of Refs. 19 and 20 are phase-based techniques, that of Ref. 21 is an energy-based method, and those of Refs. 13, 14, and 22–32 are differential techniques.

Multiframe techniques usually require the image sequence to be differentiable along the temporal axis and thus require either oversampling or large temporal support. Such techniques have been proposed to improve the accuracy of optical flow estimation, but relatively few methods have been proposed to handle situations where large numbers of consecutive frames are unavailable or the image sequences are undersampled. These situations are common in many applications. For instance, in real-time on-road commercial vehicle applications, optical flow needs to be quickly estimated as soon as a few (two or three) image frames are available to control an automatic driving system before it is too late; in many surveillance

1084-7529/2003/081505-11$15.00

systems, camera sampling rates are low (15 frames/s or lower); and many other real-time systems do not have enough computing power or memory to process large numbers of frames in real time (e.g., real-time video stabilizers). Our method provides an inexpensive way to estimate optical flow with reasonably good accuracy.

Multiscale optical flow estimation was introduced by Weber and Malik.[33] This method requires selecting a set of filters. Convolving images with many filters is relatively computationally expensive. In addition, these filters need to be chosen so that they are compatible with each other. To avoid these problems, a wavelet technique is a good candidate. Mendelsohn *et al.*[34] also proposed a multiscale coarse-to-fine method under the assumption of smooth inverse depth map instead of smooth flow field. They formulated the optical flow problem in a very different style involving a projection image system model.

Using wavelets for optical flow estimation is no longer very new (e.g., Refs. 18 and 20–22). Most previous work exploits the approximation capability of wavelets and uses them as basis functions, or exploits the phase rather than spatial information. Some of these methods require large temporal support (a long image sequence) to achieve reasonably good accuracy, or they suffer from relatively low accuracy. We use wavelets in a very different way; our approach takes advantage of the wavelet filter property and requires only two frames. We propose a low-complexity algorithm [$O(N)$] that not only is able to estimate reasonably accurate optical flow but is also suitable for hardware implementation.

2. BASIC IDEA

A. Problem Formulation

Recent research[12–14,22–32] shows that differential techniques seem more accurate in optical flow estimation. These techniques are based on the well-known constant-intensity assumption or its extended versions. The assumption is that given an image sequence $I(\mathbf{x}, t)$, the intensity changes only because of the motion of the scene, the camera, or both, i.e.,

$$I(\mathbf{x}, t) = I(\mathbf{x} - \mathbf{v}, 0), \tag{1}$$

where $\mathbf{x} = (x, y)$ is position in the image and $\mathbf{v} = (u, v)$ is optical flow. The differential version of this assumption—the gradient constraint equation—can be expressed as

$$(I_x(\mathbf{x}, t), I_y(\mathbf{x}, t)) \cdot \mathbf{v} + I_t(\mathbf{x}, t) = 0, \tag{2}$$

where $I_{x[y][t]}(\cdot)$ is the partial derivative of $I(\mathbf{x}, t)$ with respect to x, y, or t. Note that for each pixel there are two unknowns but only one equation (known as the aperture problem); more constraints are needed. In addition, because image sequences usually have aliasing problems, I_x, I_y, I_t need to be carefully handled. It is well-known that poor numerical differentiation can cause large errors.[12,22,27,35]

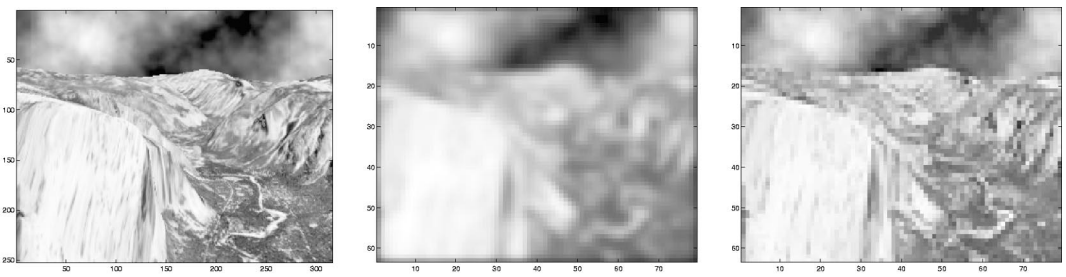
To handle the aperture problem, a locally constant motion assumption is widely used. Recent studies (e.g., Refs. 27 and 36) show that an affine motion model is more accurate because its elementary flow fields include more types of motion, such as translation, rotation, dilation, divergence, shear, and stretch. In our algorithm, we assume that the local optical flow $\mathbf{u} = (u, v)$ satisfies the following affine model:

$$u = p_1 x + p_2 y + p_3, \qquad v = p_4 x + p_5 y + p_6. \tag{3}$$

Therefore six pixels in the neighborhood can uniquely define the local optical flow. More pixels define an overdetermined linear system $\mathbf{p}$.

To handle the aliasing problem, smoothing is widely used. For spatial differentiation, since large support is always available, accurate estimates are relatively easy to obtain (e.g., by smoothing using a series-designed first-order derivative with large support[36]). Time aliasing is trickier to handle. When more than two consecutive frames are available, more methods can be applied, such as using a differentiation filter with more-than-two-frame support, using a more accurate statistical error model to solve the linear system (e.g., total least squares, extended least squares[30]), or using basis functions to approximate the image sequence and then resampling. If only two frames are available, a multiscale approach is desirable. At the coarse level, time aliasing is alleviated for velocities less than 1 pixel/frame.

The traditional multi-scale pyramid has a so-called "flattening-out" problem, which produces high errors when low-texture regions become flat at coarse levels as a result of excessive blurring. Though appropriate low-pass filtering generally improves the numerical differentiation, the excessive blurring wipes out the intensity changes and thus causes large errors in numerical differentiation, which eventually leads to poor optical flow estimation in both space and time. As an example, Figs. 1(a) and 1(b) show one frame of the Yosemite sequence



(a) (b) (c)

Fig. 1. Multiscale strategy: (a) one frame of the Yosemite sequence, (b) traditional Gaussian smoothing ($\sigma = 1.5$) and dyadic downsampling at level 2, (c) wavelet decomposition at level 2 using biorthogonal wavelets (bior1.3).

and its coarse scale after traditional Gaussian smoothing and dyadic downsampling at level 2. The former flattens out the details and gives rise to a large homogeneous area, where optical flow is hard to detect.

B. Motivation for Using Wavelets

There are several motivations for using wavelets:

- The wavelet transform has a built-in multiscale structure.
- The wavelet transform has superior spatial resolution as well as frequency resolution,[37,38] which overcomes the flattening-out problem.
- The wavelet transform separates point noise from useful signal, so that noise removal is automatically performed, to some degree, during the wavelet decomposition at each scale level.
- The biorthogonal/orthogonal and compactly supported wavelet decomposition has a fast algorithm $[O(N)]$.

Figure 1(c) shows the approximation channel of the wavelet decomposition using biorthogonal spline wavelets ("bior1.3") at level 2. Compared with the result of traditional Gaussian smoothing and downsampling, it preserves much more detail, with point noise separated into the diagonal channel (details are in Section 3).

Previously proposed multiscale approaches (e.g., Refs. 18 and 22) usually estimate the optical flow in a coarse-to-fine manner; i.e., the optical flow is first estimated at a coarse level, where time aliasing is assumed to be small, and then the result is propagated to finer levels as initial guesses. One shortcoming of this approach is that errors at the coarse level, which can be large at some pixels because of the lack of information at the coarse level, are also propagated to the finer levels. Though this can be compensated to some degree by projecting the results at the finer levels back to the coarse level and doing some adjustments, this usually results in slow or even nonconverging iteration.

Our multiscale approach is *coarse-and-fine* rather than coarse-to-fine; it estimates the optical flow at different scales simultaneously to avoid error propagation. Another alternative to coarse-to-fine was proposed by Weber and Malik,[33] where the estimate was first formed at each scale and then valid estimates were combined to yield final estimates.

Before discussing our method in detail, we will give a brief review of the wavelet transform and an important derivation.

3. HIERARCHICAL FIRST-ORDER DERIVATIVES BY WAVELET DECOMPOSITION

For the sake of clarity and simplicity, the derivation here is based on one-dimensional (1-D) continuous wavelets. The results can be easily extended to higher-dimensional discrete wavelet transforms (DWTs).

Consider the definition of the wavelet transformation:

$$(T^{\mathrm{wav}}f)(a,\,b) = \langle f,\,\psi^{a,b}\rangle = \frac{1}{\sqrt{|a|}}\int_{-\infty}^{+\infty} f(x)\overline{\psi\left(\frac{x-b}{a}\right)}\mathrm{d}x, \tag{4}$$

where $\overline{\psi(\cdot)}$ stands for the complex conjugate and $\psi(\cdot) \in L^2(R)$ is compactly supported and satisfies the admissibility condition:

$$C_\psi = \int_{-\infty}^{+\infty}\frac{|\hat{\psi}(\omega)|^2}{|\omega|}\mathrm{d}\omega < \infty. \tag{5}$$

It can be proved[37] that the admissibility condition can be satisfied only if

$$\int_{-\infty}^{+\infty}\psi(x)\mathrm{d}x = 0. \tag{6}$$

Define $h_a^0(t) = \int_{-\infty}^{t}\psi(x/a)\mathrm{d}x$. From Eq. (6), we have

$$\lim_{t\to-\infty} h_a^0(t) = 0, \qquad \lim_{t\to+\infty} h_a^0(t) = \int_{-\infty}^{+\infty}\psi\left(\frac{x}{a}\right)\mathrm{d}x = 0. \tag{7}$$

Furthermore, define $h_a^1(t)$ as follows:

$$h_a^1(t) = \int_{-\infty}^{t} h_a^0(x)\mathrm{d}x = xh_a^0(x)|_{-\infty}^{t} - \int_{-\infty}^{t} x\left[\frac{\mathrm{d}}{\mathrm{d}x}h_a^0(x)\right]\mathrm{d}x. \tag{8}$$

Recalling that $\psi(\cdot)$ is compactly supported, we have

$$\lim_{t\to-\infty} h_a^1(t) = 0,$$

$$\lim_{t\to+\infty} h_a^1(t) = -\int_{-\infty}^{+\infty} x\psi\left(\frac{x}{a}\right)\mathrm{d}x = -a^2\int_{-\infty}^{+\infty} x\psi(x)\mathrm{d}x. \tag{9}$$

For the wavelet function with n vanishing moments, repeatedly using Eq. (9), we get

$$\int_{-\infty}^{+\infty} h_a^{n-1}(x)\mathrm{d}x = (-1)^n\frac{a^{n+1}}{n!}\int_{-\infty}^{+\infty} x^n\psi(x)\mathrm{d}x \neq 0. \tag{10}$$

Thus, for the wavelet function $\psi(x) \in L^2(R)$ with n vanishing moments, we define a function

$$h(t) = c\int_{-\infty}^{t}\int_{-\infty}^{t_1}\cdots\int_{-\infty}^{t_{n-2}}\psi(x)\mathrm{d}x\mathrm{d}t_{n-2}\cdots\mathrm{d}t_1\mathrm{d}t, \tag{11}$$

where c is a constant, so that

$$\int_{-\infty}^{+\infty} h(x)\mathrm{d}x = 1. \tag{12}$$

Equations (5) and (6) guarantee that $h(t)$ also satisfies

$$\lim_{t\to-\infty} h(t) = \lim_{t\to+\infty} h(t) = 0. \tag{13}$$

We call a function that satisfies Eqs. (12) and (13) a smoothing function.

In the case of a real signal, we have

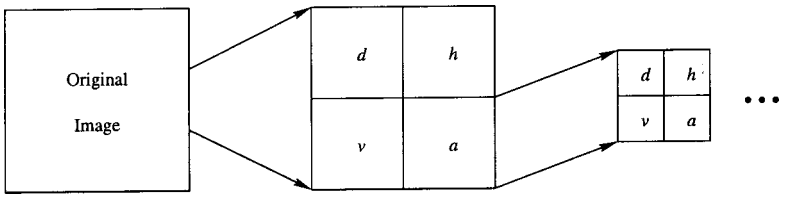


Fig. 2. Wavelet decomposition: a, approximation channel; h, v, horizontal and vertical channels; d, diagonal channel.

$$
\begin{aligned}
(T^{\mathrm{wav}}f)(a, b) &= \langle f, \psi^{a,b}\rangle \\
&= \frac{1}{\sqrt{|a|}}\int_{-\infty}^{+\infty} f(x)\psi\left(\frac{x-b}{a}\right)\mathrm{d}x \\
&= (-1)^n\frac{c}{\sqrt{|a|}}\int_{-\infty}^{+\infty} f(x)\left[\frac{\mathrm{d}^n}{\mathrm{d}x^n}h\left(\frac{x-b}{a}\right)\right]\mathrm{d}x \\
&= (-1)^n\frac{c}{\sqrt{|a|}}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\int_{-\infty}^{+\infty} f(x)h\left(\frac{x-b}{a}\right)\mathrm{d}x.
\end{aligned}
\tag{14}
$$

That is, the wavelet transform obtained by using the compactly supported wavelet function $\psi(x) \in L^2(R)$ with n vanishing moments can be regarded as the nth derivative of the signal after a scaled smoothing operation is performed. When $n = 1$, we get the hierarchical first derivatives that we need.

In the discrete case, the coefficients of the wavelet expansion of a signal constitute the DWT.[37] The above result holds for the DWT if we replace integration with summation and differentiation with differencing. The 1-D DWT can be easily extended to two-dimensional by tensor product[37]:

$$
\begin{aligned}
\psi^a(x, y) &= \phi(x)\phi(y), \qquad \psi^h(x, y) = \phi(x)\psi(y), \\
\psi^v(x, y) &= \psi(x)\phi(y), \qquad \psi^d(x, y) = \psi(x)\psi(y),
\end{aligned}
\tag{15}
$$

where $\psi^a(x, y)$ is the two-dimensional (2-D) scaling function and $\psi^{h,v,d}(x, y)$ are the corresponding 2-D wavelets. Applying these functions (whose compactly supported wavelet base has n vanishing moments) to a 2-D signal, we get four channels of decomposed signals:

- Channel a: the approximation channel, a coarse-scale version of the original signal.
- Channels h and v: the horizontal and vertical channels, which are the nth derivatives (when using a compactly supported wavelet base with n vanishing moments) of the original signal in the two orientations after a smoothing operation. These two channels can be combined to get nth-derivative vectors.
- Channel d: the diagonal channel, which includes nth "corners" in the original signal.

These operations can be applied to channel a recursively to get a hierarchical signal representation (see Fig. 2).

The above derivation gives an important result: If we carefully select a compactly supported wavelet base with one vanishing moment, e.g., "Haar" wavelets, we get a hierarchical signal and its first-order derivatives from the wavelet decomposition. The superior frequency and spatial resolutions of the wavelet transform[37,38] guarantee that details are largely preserved at the coarse level, so that the first-order derivatives at each scale are accurate. Thus we obtain the hierarchical gradient constraint equations (2) without a serious flattening-out problem. Here the hierarchical time derivatives (I_t's), which have less time aliasing at the coarse level, can be easily calculated by taking first-order forward differences at each scale.

4. COARSE-AND-FINE OPTICAL FLOW ESTIMATION

Our method is formulated in a coarse-and-fine manner to avoid the error propagation problem in the traditional coarse-to-fine approach. Let the superscript (e.g., l) denote the level number (the original image is at level $l = 0$), the first subscript (e.g., x, y, and t) denote the direction of the first-order derivative or wavelet decomposition channel (e.g., a, h, v, and d), and the second subscript (e.g., k) denote the frame number. For example, $I^l_{x[\,y\,][t],k}$ is the first-order derivative of the kth frame image intensity in the $x[\,y\,][t]$ direction at scale level l, $I_{a,k}$ is the approximation channel of the kth frame, and $I_{d,k}$ is the diagonal channel of the kth frame. The wavelet filter unit is designed as in Fig. 3. The symbol "2↓" denotes dyadic downsampling. By selecting the compactly supported wavelet basis with one vanishing moment, we can get the hierarchical gradient constraint equations (Fig. 4) with accurate first-order derivatives at each level.

Because of the dyadic downsampling during the wavelet decomposition, 1 pixel at the coarse level represents 2×2 pixels at the next-finer level. Thus the optical flow of the pixel at the coarse level is half of the average of the optical flows of the 2×2 pixels at the finer level. In general, 1 pixel at the coarsest level represents $2^{L-l} \times 2^{L-l}$ pixels at levels $l = 0, 1, \ldots, L$, where L is the maximum degree of wavelet decomposition. Instead of estimating the optical flow at the coarsest level and then propagating the result to the finer level, our method estimates the optical flow at all scale levels simultaneously.

Mathematically, in the $m \times n$ neighborhood S of the pixel (x_i, y_i) in the original image, Eqs. (3) can be rewritten as

$$
\mathbf{u} = f(\mathbf{X}, \mathbf{Y})\mathbf{p}, \tag{16}
$$

where $\mathbf{X}$ and $\mathbf{Y}$ are x- and y-coordinate matrices and $\mathbf{p} = (p_1, p_2, p_3, p_4, p_5, p_6)^{\mathrm{T}}$ is the vector of affine model parameters. Function $f(\cdot, \cdot)$ is defined as follows:

$$f(\mathbf{X}, \mathbf{Y}) = \begin{bmatrix} \cdots & & & \cdots & & \\ x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 \\ & \cdots & & & \cdots & \end{bmatrix}^{\mathrm{T}}, \tag{17}$$

where $(x_i, y_i) \in S$, $i = 1, 2, \ldots, M$, and $M = m \times n$. At level l, $m = n = 2^{L-l}$.

Substituting Eq. (16) into Eq. (2), we have

$$\mathbf{I_x} \cdot f(\mathbf{X}, \mathbf{Y})\mathbf{p} = -\mathbf{I_t}, \tag{18}$$

where

$$\mathbf{I_x} = \begin{bmatrix} I_{x_1} & I_{y_1} & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & I_{x_M} & I_{y_M} \end{bmatrix}, \tag{19}$$

$$\mathbf{I_t} = (I_{t_1}, I_{t_2}, \ldots, I_{t_M})^{\mathrm{T}}. \tag{20}$$

Let $D(\cdot)$ be the dyadic downsampling function and $\mathbf{W}$ be the averaging operator. Recalling that the superscript l denotes the level number, we have the hierarchical gradient constraint functions

$$\mathbf{A}^l\mathbf{p} = -\mathbf{I}_\mathbf{t}^l, \tag{21}$$

where

$$\mathbf{A}^l = \mathbf{I}_\mathbf{x}^l \cdot f(\mathbf{X}^l, \mathbf{Y}^l). \tag{22}$$

Here $\mathbf{X}^l$ and $\mathbf{Y}^l$ are iteratively defined by

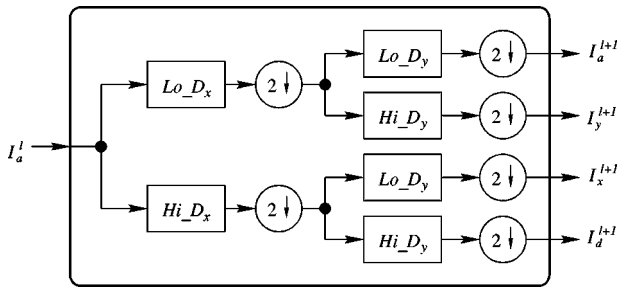


Fig. 3. Wavelet filter unit, where $Lo\_D_{x(y)}$ is the decomposition low-pass filter and $Hi\_D_{x(y)}$ is the decomposition high-pass filter. See the text for details.

$$\mathbf{X}^l = D(\mathbf{W} \cdot \mathbf{X}^{l-1}), \qquad \mathbf{X}^0 = \mathbf{X},$$
$$\mathbf{Y}^l = D(\mathbf{W} \cdot \mathbf{Y}^{l-1}), \qquad \mathbf{Y}^0 = \mathbf{Y}. \tag{23}$$

Therefore each pixel at level l corresponds to 2×2 pixels at level $l - 1$. For L degrees of wavelet decomposition, this corresponds to $2^L \times 2^L$ pixels in the original images. Their affine parameters can be estimated by the following linear equations, integrating the gradient constraint at all scale levels:

$$\mathbf{Ap} = \mathbf{b}, \tag{24}$$

where

$$\mathbf{A} = [\mathbf{A}^0, \mathbf{A}^1, \ldots, \mathbf{A}^L]^{\mathrm{T}}, \qquad \mathbf{b} = -(\mathbf{I}_\mathbf{t}^0, \mathbf{I}_\mathbf{t}^1, \ldots, \mathbf{I}_\mathbf{t}^L)^{\mathrm{T}}. \tag{25}$$

When the wavelet basis is symmetric, the alignment of levels is trivial. Once the affine model parameter $\mathbf{p}$ is computed by solving Eq. (24) in the sense of least squares, the optical flow of the central 2×2 pixels of the $2^L \times 2^L$ neighborhood in the original image can be estimated by using Eqs. (3). This is illustrated in Fig. 5.

5. ALGORITHM COMPLEXITY ANALYSIS

A. Algorithm

The algorithm for fast optical flow estimation is summarized below (the framework of the algorithm is illustrated in Fig. 5):

FAST_OF_ESTIMATION(I_1, I_2)

I_1: Image frame t.

I_2: Image frame $t + 1$.

The following two steps need to be precomputed only once, and the results are stored for future use:

- Preconstruct matrices $f(\mathbf{X}^l, \mathbf{Y}^l)$, $l = 0,1,\ldots,L$, from Eq. (17).
- Preconstruct matrix $f(\mathbf{X}, \mathbf{Y}) = [f(\mathbf{X}^0, \mathbf{Y}^0), f(\mathbf{X}^1, \mathbf{Y}^1), \ldots, f(\mathbf{X}^L, \mathbf{Y}^L)]^{\mathrm{T}}$.

1. $I_{a,1}^0 \leftarrow I_1$, $I_{a,2}^0 \leftarrow I_2$.
2. $L \leftarrow$ maximum number of decompositions. The determination of L is discussed in Subsection 5.B.
3. Decompose the two consecutive image frames I_1, I_2, by using some compactly supported wavelet basis with

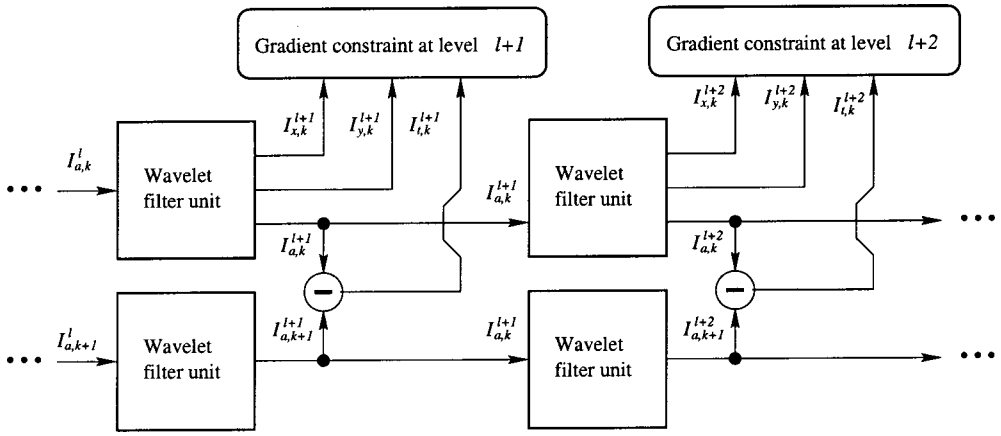


Fig. 4. Hierarchical coarse-and-fine gradient constraint.

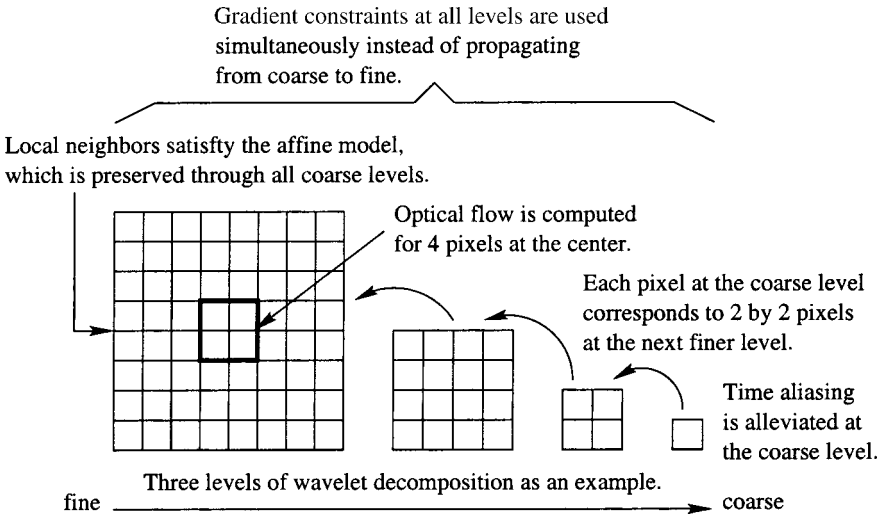


Fig. 5. Hierarchical coarse-and-fine optical flow estimation.

one vanishing moment (e.g., biorthogonal spline wavelets: bior1.3) up to level L to get hierarchical images $I^l_{a,1}$, $I^l_{a,2}$ and their first-order spatial derivatives $I^l_{x,1}$, $I^l_{y,1}$, $l = 1, 2,..., L$. This can be done by the fast algorithm described in Fig. 3 and Ref. 38.

4. Compute the first-order spatial derivatives $I^0_{x,1}$, $I^0_{y,1}$ of the original images ($l = 0$).

5. Compute the first-order temporal derivatives at each level l by forward differentiation: $I^l_{t,1} = I^l_{a,2} - I^l_{a,1}$, $l = 0, 1,..., L$.

6. **for** each 2×2 pixels **do**

7. Construct spatial derivative matrices $\mathbf{I}^l_{\mathbf{x}}$ [Eq. (19)] and temporal derivative vectors $\mathbf{I}^l_{\mathbf{t}}$ [Eq. (20)] for each level $l = 0, 1, 2,..., L$.

8. Compute matrix $\mathbf{A}^l$ from Eq. (22).

9. Construct the linear system in Eq. (24) and solve it in the sense of least squares to get the affine model parameters $\mathbf{p}$.

10. Compute the optical flow by using Eqs. (3).

11. **end for**

12. Output the optical flow.

B. Algorithm Complexity

The value of the maximum number of wavelet levels L is determined by both the motion model and the maximum optical flow magnitude. On the one hand, linear system (24) has $(4^{L+1} - 1)/3$ equations and six unknowns; this requires $(4^{L+1} - 1)/3 \geqslant 6 \Rightarrow L \geqslant 2$. On the other hand, if the maximum optical flow magnitude $V_{\max}$ can be estimated, it can help to determine the upper bound of L. At level

$$L = [\log_2 V_{\max}], \tag{26}$$

the optical flow is already less than 1 pixel/frame (and thus has less time aliasing). Usually $L \leqslant 5$; too deep a decomposition causes the coarsest level to contain too little useful information.

Table 1. Algorithm Efficiency with bior1.3 (in Flops per Pixel)

L	Wavelet Transform	Construct Matrix $\mathbf{A}$	Solve Linear Equations	Compute Affine Optical Flow	Total
2	36		207		255
3	42	4	783	8	837
4	45		3089		3144
Magarey's algorithm (version 1, faster)					234
Magarey's algorithm (version 2, more accurate)					1618
Bernard's algorithm (with illumination)					863
Bernard's algorithm (without illumination)					780

Assume that one frame has N pixels. It is then obvious from the wavelet transform structure that the complexity is $O(N)$. The following is the algorithm efficiency in terms of flops per pixel, assuming that the standard normal-equations method[39] is used to give the least-squares solution for the linear system:

$$\text{flops} = \underbrace{4D\left(2 - \frac{1}{2^{L-1}}\right)}_{\substack{\text{wavelet}\\ \text{decomposition}}} + \underbrace{4}_{\substack{\text{construct}\\ \text{matrix } \mathbf{A}}} + \underbrace{3 \times 4^{L+1} + 15}_{\substack{\text{solve linear}\\ \text{equations}}} + \underbrace{8}_{\substack{\text{compute affine}\\ \text{optical flow}}}, \tag{27}$$

where D is the length of the wavelet filters. Table 1 lists the algorithm efficiency for the compactly supported biorthogonal spline wavelet filters bior1.3, whose filter length is $D = 6$. As a comparison, the performance of Magarey and Kingsburg's efficient algorithm[20] is also given. It can be observed that most of the flops are consumed by solving the linear equations.

6. EXPERIMENTS

Compactly supported biorthogonal spline wavelet filters of order $N_r = 1$ and $N_d = 3$ (denoted by bior1.3) were used throughout the experiments because they have one vanishing moment and are symmetric (so that the shift problem is avoided):

$$f_{\text{low}} = \frac{\sqrt{2}}{16}(-1 \quad 1 \quad 8 \quad 8 \quad 1 \quad -1),$$
$$f_{\text{high}} = \frac{\sqrt{2}}{2}(0 \quad 0 \quad -1 \quad 1 \quad 0 \quad 0). \tag{28}$$

Unless stated otherwise, all images were spatially

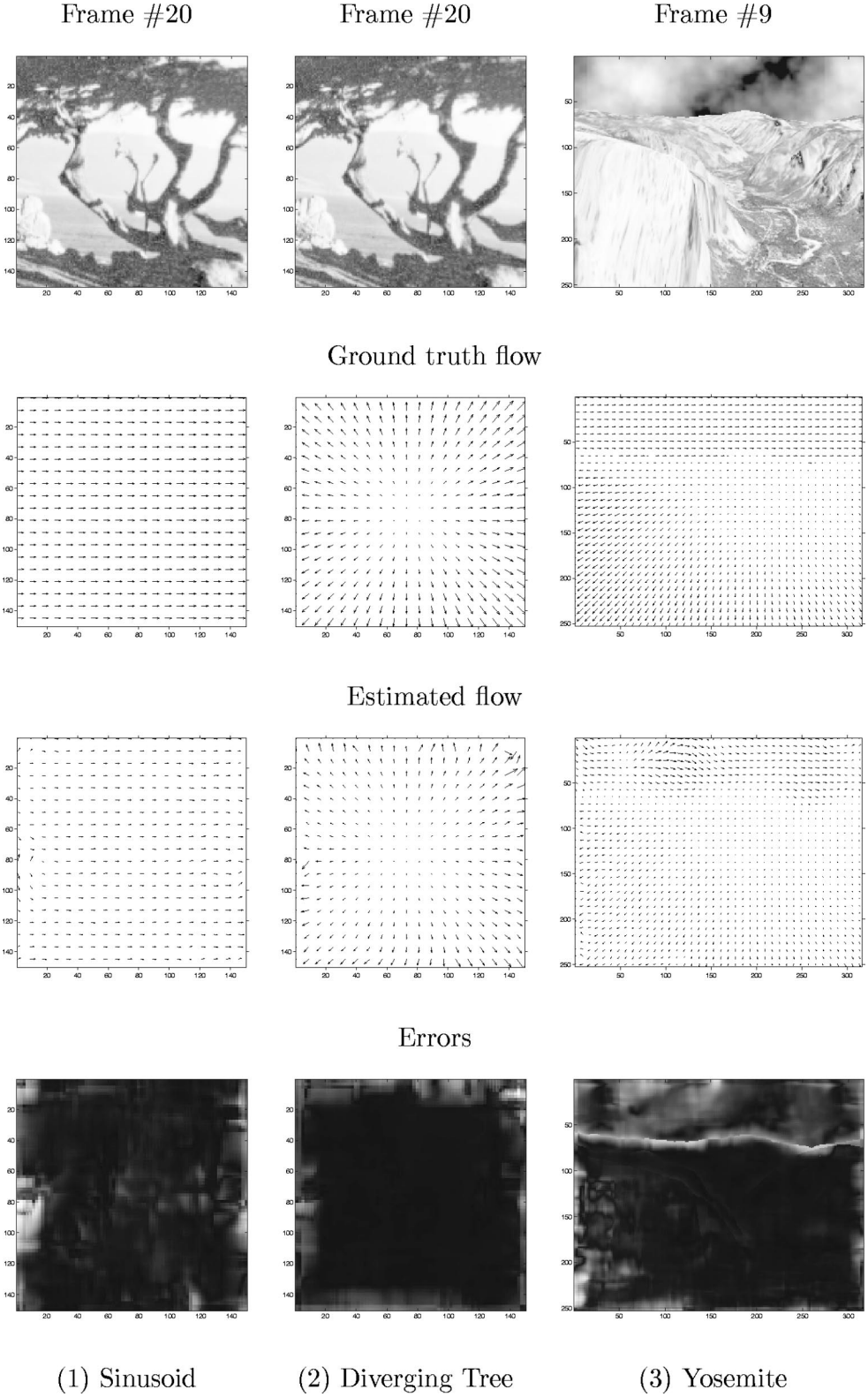


Fig. 6. Synthetic image sequence: row 1, one frame from each sequence; row 2, ground-truth optical flow; row 3, optical flow estimated by proposed method; row 4, error distribution.

Table 2. Translating Tree Results (Frame 20)

Technique	Frames Used	Average Error	Standard Deviation	Density (%)
Our algorithm	2	3.67°	2.18°	100
Horn and Schunck (original)	2	38.72°	27.67°	100
Anandan	2	4.54°	3.10°	100
Singh ($n = 2$, ω=2)	2	1.25°	3.29°	100
Heeger (level 0)	2	8.10°	12.30°	77.9
Heeger (level 1)	2	4.53°	2.14°	57.8
Magarey (version 1, faster)	2	2.31°	—	100
Magarey (version 2, more accurate)	2	1.32°	—	100
Bernard	2	0.78°	—	99.30
Srinivasen and Chellappa	2	0.61°	0.26°	100
Horn and Schunck (modified)	7–13	2.02°	2.27°	100
Nagel	7–24	2.44°	3.06°	100
Liu *et al.*	11	0.66°	0.83°	100
Weber and Malik	10	0.49°	0.35°	96.8

Table 3. Diverging Tree Results (Frame 20)

Technique	Frames Used	Average Error	Standard Deviation	Density (%)
Our algorithm	2	1.67°	0.88°	100
Horn and Schunck (original)	2	12.02°	11.72°	100
Anandan	2	7.64°	4.96	100
Singh (step 2, $n = 2$, $\omega = 2$)	2	8.60°	4.78°	100
Heeger (level 0)	2	4.95°	3.09°	73.8
Magarey (version 1, faster)	2	3.92°	—	100
Magarey (version 2, more accurate)	2	3.12°	—	100
Bernard	2	~1.2 times higher than Heeger, etc.		
Srinivasen and Chellappa	2	2.94°	1.64	100
Horn and Schunck (modified)	7–13	2.55°	3.67°	100
Nagel	7–24	2.94°	3.23°	100
Liu *et al.*	11	1.86°	1.35°	100
Weber and Malik	10	3.18°	2.50°	88.6

Table 4. Yosemite Results (Frame 9)

Technique	Frames Used	Average Error	Standard Deviation	Density (%)
Our algorithm	2	8.43°	10.12°	100
Horn and Schunck (original)	2	32.43°	30.28°	100
Anandan	2	15.84°	13.46°	100
Singh (step 2, $n = 2$, ω=2)	2	13.16°	12.07°	100
Heeger (level 0)	2	20.89°	34.26°	64.2
Magarey (version 1, faster)	2	7.70°	—	100
Magarey (version 2, more accurate)	2	6.20°	—	100
Bernard	2	6.5°	—	96.50
Srinivasen and Chellappa	2	8.94°	10.63°	100
Horn and Schunck (modified)	7–13	11.26°	16.41°	100
Nagel	7–24	11.71°	10.59°	100
Liu *et al.*	11	7.52°	13.72°	100
Weber and Malik	10	4.31°	8.66°	64.2

smoothed by the Gaussian filter with standard deviation $\sigma = 2$ before further processing. The total number of wavelet decompositions for each sequence is 5. The first-order derivatives of the images at level 0 (the original images) are computed through the following series-designed filter:

$$\frac{1}{2520}(-2 \quad 25 \quad -150 \quad 600 \quad -2100 \quad 0 \quad 2100 \quad -600 \quad 150 \quad -25 \quad 2). \quad (29)$$

In the computation of errors, pixels out of the filter support were not counted, since derivatives at such pixels were not accurate anyway.

A. Synthetic Image Sequences

Three synthetic image sequences were used to test our algorithm quantitatively and compare it with other optical flow techniques:

- Translating Tree [Fig. 6(1)]. This sequence simulates translational camera motion along the X axis parallel to a textured planar surface. The velocities are between 1.73 and 2.26 pixels/frame.
- Diverging Tree [Fig. 6(2)]. This sequence simulates a synthetic camera moving toward a planar image of a tree. The speeds range from 0 in the middle (at the focus of expansion) to 1.4 pixels/frame on the left and 2.0 pixels/frame on the right.
- Yosemite [Fig. 6(3)]. This is a more complex test sequence with a wide range of velocities, occluding edges, and severe aliasing in the lower portions of the images.

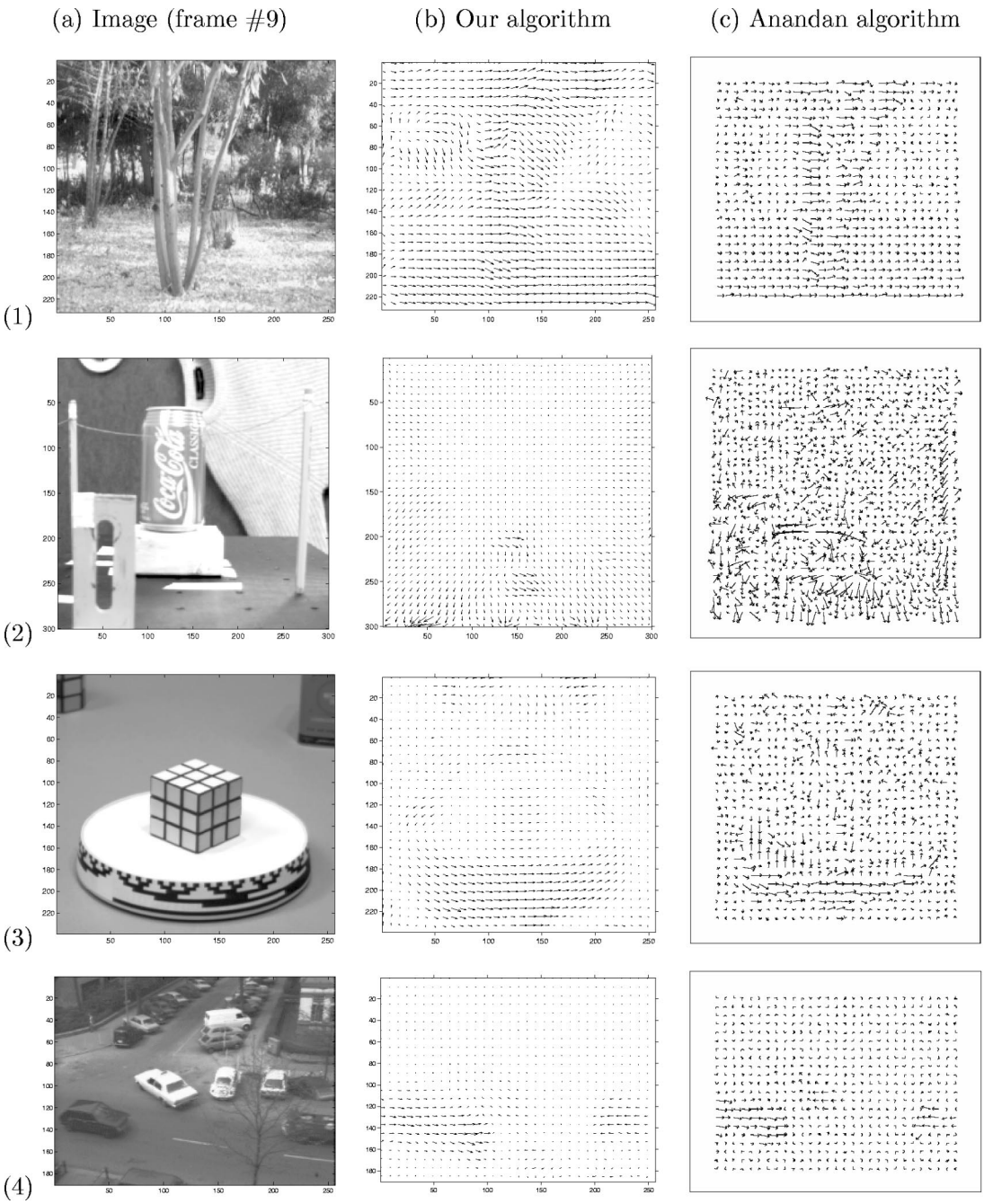


Fig. 7. Real sequences: (1) SRI Trees, (2) NASA, (3) Rubik Cube, (4) Hamburg Taxi.

The clouds translate to the right at a speed of 2 pixels/frame with illumination changes. The velocity of the lower-left part is approximately 4 pixels/frame.

Following Refs. 12 and 40, the angular error between the correct flow $\mathbf{v}_c = (v_{c_x}, v_{c_y}, 1)^{\mathrm{T}}$ and an estimate $\mathbf{v}_e = (v_{e_x}, v_{e_y}, 1)^{\mathrm{T}}$ is calculated:

$$\text{error} = \arccos\left(\frac{\mathbf{v}_c}{\|\mathbf{v}_c\|_2} \cdot \frac{\mathbf{v}_e}{\|\mathbf{v}_e\|_2}\right). \tag{30}$$

The advantage of using the angular error is discussed in Ref. 12.

Results are shown in Fig. 6. The ground-truth flow and the error distribution are also shown. It can be observed that errors mainly occur on the boundary of the scene. For the Yosemite sequence, errors also occur where the illumination changes (the cloud area) and the flow has large discontinuities. Error statistics, including mean and standard deviation, are listed in Tables 2–4. For comparison, results from other two-frame methods in Refs. 12 and 22 are given. Some multiframe dense optical flow results in Refs. 12 and 27 are also given.

In general, our flow estimates are accurate and have high density. For these image sequences, our algorithm outperforms most two-frame techniques listed (except Singh's method for the Translating Tree sequence), and even some of the multiframe methods in Refs. 12 and 27 for the Diverging Tree and Yosemite sequences, as regards both flow accuracy and density.

B. Real Image Sequences

For comparison with other optical flow techniques,[12,18,22,24,27] four widely used image sequences were selected to test our algorithm:

- SRI Trees (translation dominated); see Fig. 7(1). In this sequence, the camera translates parallel to the ground plane. This sequence is very challenging because of poor resolution, considerable occlusion, and low contrast.[12] The maximum velocity is approximately 2 pixels/frame.
- NASA (dilation dominated); see Fig. 7(2). This sequence was obtained by moving the camera along the line of sight toward the cola can near the center of the image.
- Rubik Cube (rotation dominated); see Fig. 7(3). The Rubik cube is rotating counterclockwise on a turntable.
- Hamburg Taxi (translation and rotation); see Fig. 7(4). There are three obvious objects moving independently: the taxi turning at the corner, a car at the lower left driving from left to right, and a van at the lower right driving from right to left.

As a comparison, we show flow results for both our algorithm and the Anandan algorithm.[12] The Anandan algorithm is a two-frame Laplacian pyramid and a coarse-to-fine sum-of-squared-difference-based method, while our algorithm is a two-frame wavelet pyramid and coarse-and-fine differentiation-based approach.

The overall results obtained from our algorithm are very reasonable and appear to be much more accurate than the results of the two-frame techniques in Ref. 12 and other recently published papers (e.g., Ref. 22). Even compared with the multiframe techniques in Ref. 12 and some other recently published papers (e.g., Refs. 18, 24, and 27), our results appear to be better or at least no worse. Even in sequences with very large rotation, dilation, and shadow, our algorithm still produces very reasonable optical flow estimates.

7. CONCLUSIONS

We have presented a wavelet decomposition approach to compute multiscale optical flow by using only two image frames. Our algorithm is based on the fact that if a compactly supported wavelet basis with one vanishing moment is carefully selected, we can get hierarchical images, first-order derivatives, and corners from the wavelet decomposition. With the superior space as well as frequency resolution of the wavelet filters, the "flattening-out" problem is well handled. The differential technique is thus formulated in a hierarchical framework with an affine model. The aperture and time-aliasing problems are circumvented in the multiscale hierarchy. Optical flow is estimated in a coarse-and-fine manner, which overcomes the error propagation problem in the traditional coarse-to-fine method. Experiments demonstrate that our algorithm is accurate and robust. The two-frame estimate and the relatively low computational complexity of our algorithm could be exploited for real-time accurate optical flow estimation.

ACKNOWLEDGMENT

This work was partially supported by U.S. Office of Naval Research grant N00014-01-1-0265.

H. Liu's e-mail address is hyliu@cfar.umd.edu.

REFERENCES

1. J. Gibson, *The Senses Considered as Perceptual Systems* (Houghton Mifflin, Boston, 1966).
2. J. Oliensis, "A critique of structure-from-motion algorithms," Comput. Vis. Image Underst. **80**, 172–214 (2000).
3. G. J. Young and R. Chellappa, "3-D motion estimation using a sequence of noisy stereo images: models, estimation and uniqueness results," IEEE Trans. Pattern Anal. Mach. Intell. **14**, 995–1013 (1992).
4. J. Duncan and T. Chou, "On the detection of motion and the computation of optical flow," IEEE Trans. Pattern Anal. Mach. Intell. **14**, 346–352 (1992).
5. K. Prazdny, "Determining the instantaneous direction of motion from optical flow generated by a curvilinearly moving observer," Comput. Graph. Image Process. **17**, 82–87 (1981).
6. Y. Aloimonos and Z. Duric, "Estimating the heading direction using normal flow," Int. J. Comput. Vision **13**, 33–56 (1994).
7. M. Hashimoto and J. Sklansky, "Multiple-order derivatives for detecting local image characteristics," Comput. Vision Image Understand. **39**, 28–55 (1987).
8. W. Thompson, K. Mutch, and V. Berzins, "Dynamic occlusion analysis in optical flow fields," IEEE Trans. Pattern Anal. Mach. Intell. **7**, 374–383 (1985).
9. W.-S. Chou and Y.-C. Chen, "Estimation of the velocity field of two-dimensional deformable motion," Pattern Recogn. **26**, 351–364 (1993).
10. J. Gibson, "Optical motions and transformations as stimuli for visual perception," Psychol. Rev. **64**, 288–295 (1957).

11. S. Ullman, "The interpretation of structure from motion," Proc. R. Soc. London Ser. B **203**, 405–426 (1979).
12. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," Int. J. Comput. Vision **12**, 43–77 (1994).
13. P. J. Sobey and M. V. Srinivasan, "Measurement of complex optical flow with use of an augmented generalized gradient scheme," J. Opt. Soc. Am. A **11**, 2787–2798 (1994).
14. P. J. Sobey, M. G. Nagle, Y. V. Venkatesh, and M. V. Srinivasan, "Measurement of optical flow by a generalized gradient scheme," J. Opt. Soc. Am. A **8**, 1488–1498 (1991).
15. A. G. Borş and I. Pitas, "Optical flow estimation and moving object segmentation based on median radial basis function network," IEEE Trans. Image Process. **7**, 693–702 (1998).
16. E. Meminémin and P. Pérez, "Dense estimation and object-based segmentation of the optical flow with robust techniques," IEEE Trans. Image Process. **7**, 703–719 (1998).
17. J. N. Pan, Y. Q. Shi, and C. Q. Shu, "Correlation-feedback technique in optical flow determination," IEEE Trans. Image Process. **7**, 1061–1067 (1998).
18. Y. T. Wu, T. Kanade, J. Cohn, and C. C. Li, "Optical flow estimation using wavelet motion model," in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE Computer Society Press, Los Alamitos, Calif., 1998), pp. 992–998.
19. M. Hemmendorff, M. T. Andersson, and H. Knutsson, "Phase-based image motion estimation and registration," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Institute of Electrical and Electronics Engineers, New York, 1999), pp. 3345–3348.
20. J. Magarey and N. Kingsbury, "Motion estimation using a complex-value wavelet transform," IEEE Trans. Signal Process. **46**, 1069–1084 (1998).
21. F. A. Mujica, J. P. Leduc, R. Murenzi, and M. J. T. Smith, "A new motion parameter estimation algorithm based on the continuous wavelet transform," IEEE Trans. Image Process. **9**, 873–888 (2000).
22. C. P. Bernard, "Discrete wavelet analysis: a new framework for fast optical flow computation," in *Proceedings of the European Conference on Computer Vision* (Springer, Freiburg, Germany, 1998), pp. 354–368.
23. A. D. Bimbo, P. Nesi, and J. L. C. Sanz, "Optical flow computation using extended constraints," IEEE Trans. Image Process. **5**, 720–739 (1996).
24. S. Ghosal and P. Vaněk, "A fast scalable algorithm for discontinuous optical flow estimation," IEEE Trans. Pattern Anal. Mach. Intell. **18**, 181–194 (1996).
25. H. W. Haussecker and D. J. Fleet, "Computing optical flow with physical models of brightness variation," IEEE Trans. Pattern Anal. Mach. Intell. **23**, 661–673 (2001).
26. A. Kumar, A. R. Tannenbaum, and G. J. Balas, "Optical flow: a curve evolution approach," IEEE Trans. Image Process. **5**, 598–610 (1996).
27. H. Liu, T. H. Hong, M. Herman, and R. Chellappa, "A general motion model and spatio-temporal filters for computing optical flow," Int. J. Comput. Vis. **22**, 141–172 (1997).
28. S. Rakshit and C. H. Anderson, "Computation of optical flow using basis functions," IEEE Trans. Image Process. **6**, 1246–1254 (1997).
29. H. Spies and H. Scharr, "Acurate optical flow in noisy image sequences," in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE Computer Society Press, Los Alamitos, Calif., 2001), pp. 587–592.
30. S. Srinivasan and R. Chellappa, "Noise-resilient estimation of optical flow by use of overlapped basis functions," J. Opt. Soc. Am. A **16**, 493–507 (1999).
31. M. Tistarelli, "Multiple constraints to compute optical flow," IEEE Trans. Pattern Anal. Mach. Intell. **18**, 1243–1250 (1996).
32. M. Yeasin, "Optical flow in log-mapped image plane—a new approach," IEEE Trans. Pattern Anal. Mach. Intell. **24**, 125–131 (2002).
33. J. Weber and J. Malik, "Robust computation of optical flow in a multi-scale differential framework," Int. J. Comput. Vision **14**, 67–81 (1995).
34. J. Mendelsohn, E. Simoncelli, and R. Bajcsy, "Discrete-time rigidity-constrained optical flow," in *Proceedings of the International Conference on Computer Analysis of Images and Patterns* (IEEE Computer Society Press, Los Alamitos, Calif., 1997), pp. 255–262.
35. H. Liu, T. Hong, M. Herman, T. Camus, and R. Chellapa, "Accuracy vs. efficiency trade-off in optical flow algorithms," Comput. Vision Image Understand. **72**, 271–286 (1998).
36. H. Haussecker and H. Spies, *Handbook of Computer Vision and Applications Volume 2: Signal Processing and Pattern Recognition* (New York, Academic Press, 1999), chap. 13: Motion, pp. 309–396.
37. I. Daubechies, *Ten Lectures on Wavelets* (Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1992).
38. S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," IEEE Trans. Pattern Anal. Mach. Intell. **11**, 674–693 (1989).
39. G. H. Golub and C. F. V. Loan, *Matrix Computations* (Johns Hopkins U. Press, Baltimore, Md., 1996).
40. D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," Int. J. Comput. Vision **5**, 77–104 (1990).