A Three-Frame Algorithm for Estimating Two-Component Image Motion

James R. Bergen, Peter J. Burt, *Member, IEEE*, Rajesh Hingorani, and Shmuel Peleg, *Member, IEEE*

***Abstract*— A fundamental assumption made in formulating optical-flow algorithms is that motion at any point in an image can be represented as a single pattern component undergoing a simple translation; even complex motion will appear as a uniform displacement when viewed through a sufficiently small window.**

This assumption fails for a number of situations that commonly occur in real-world images. For example, transparent surfaces moving past one another yield two motion components at a point. More important, it fails along the boundary between two differently moving image regions. Even local motion analysis must be performed within a window of finite size. This window contains two motion components when it falls on a motion boundary.

We propose an alternative formulation of the local motion assumption in which there may be two distinct patterns undergoing coherent (e.g., affine) motion within a given local analysis region. We then present an algorithm for the analysis of two-component motion in which tracking and nulling mechanisms applied to three consecutive image frames separate and estimate the individual components. Precise results are obtained even for components that differ only slightly in velocity as well as for a faint component in the presence of a dominant, masking component.

We demonstrate that the algorithm provides precise motion estimates for a set of elementary two-motion configurations and show that it is robust in the presence of noise.

I. Introduction

The OPTICAL flow approach to motion analysis has been based on a *single-component model* of local image motion; even a complex moving scene will be indistinguishable from a single pattern undergoing simple translation when viewed through a sufficiently small window over a sufficiently short interval of time. Therefore, in attempting to solve the optical flow equation, it is frequently assumed that the image pattern in the immediate neighborhood of each sample point of an image sequence undergoes simple translation between image frames [7], [12], [16]. However, a single-component motion model is inadequate for a number of important situations that commonly occur in real-world image sequences. For example, transparent surfaces moving past one another yield two motion components at a point. Patterns of light and shadow moving over a differently moving surface also yield two motions. Furthermore, failures of the single-motion model occur along the boundary between any two differently moving regions in a scene. The area subject to such failures can represent a significant fraction of the area of a scene. These failures result from the fact that neighborhoods used in estimating motion cannot always be "sufficiently small." The neighborhood must be large compared with the frame-to-frame image displacements and sufficiently large to encompass adequate pattern detail on which to base estimates of motion. When this neighborhood falls on a motion boundary, the estimated motion typically represents an average of the components on either side of the boundary. It does not represent either motion accurately.

The single component model is implicit in the "smoothness constraints" used in optical flow computation [2], [11], [13]. In an effort to increase accuracy near boundaries, more recent approaches have adopted a *piecewise smoothness constraint*, which allows a small number of discontinuities between smoothly varying regions. In effect, a segmentation process is introduced to locate motion boundaries. Motion analysis is then constrained not to combine local estimates across such boundaries. However, such segmentation presents its own problems. Often, the only information on which to base segmentation is the observed image motion itself. Thus, good quality motion analysis depends on image segmentation, whereas segmentation depends in turn on good quality motion information. Methods can be readily imagined, some of which have been implemented, that alternate between computation of motion and computation of image segments that rely on successive refinement to converge to a stable interpretation of the scene [18], [20]. Examples of this approach include Markov random field models incorporating "line processes" to decouple motion estimation processes on either side of a boundary and "brittle membrane" models [8]. These techniques tend to be slow to converge and are cumbersome to apply to practical problems. In addition, segmentation-based techniques cannot deal with other types of multiple motion such as transparency.

Hough transform and correlation techniques have been used to estimate multiple components of motion without segmentation [7], [9]. A direct estimation technique has also been proposed [21]. These techniques have limited precision, however. Since the differently moving pattern components are not isolated, each component can introduce errors in the estimates obtained for the other components. It has been demonstrated [1] that rigid motions of multiple moving objects can be computed from accurate optical flow. However, traditional methods to compute optical flow fail in the case of

Manuscript received January 19, 1990; revised February 17, 1992. Recommended for acceptance by Associate Editor N. Ahuja.

J. R. Bergen and P. J. Burt are with the David Sarnoff Research Center, Subsidiary of SRI International, Princeton, NJ 08543-5300.

R. Hingorani is with the HDTV Systems Development Department, AT&T Bell Laboratories, Murray Hill, NJ.

S. Peleg is with the Department of Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel.

IEEE Log Number 9201773.

multiple moving objects.

In this paper, we introduce an alternative model for describing *local* motion in an image in which there may be two distinct, differently moving patterns within the neighborhood of an image point. We further define an algorithm that can obtain precise estimates of the component motions without explicit segmentation. This *two-component motion model* allows analysis of most basic local motion configurations that do not conform to the traditional single-motion model. The algorithm is iterative, alternately estimating one component and then the other. As each component is estimated, it is largely removed from the image through a nulling procedure. This allows more precise estimation of the remaining component. Because we relax the single-motion constraint, analysis can be performed within larger neighborhoods. This improves signal/noise aspects of the computation and leads to more precise and robust motion estimates. Convergence is rapid; in our experience, estimates of both motions are recovered to an accuracy of 0.01 pixel/frame interval after only a few iterations. The algorithm uses three frames of a motion sequence to estimate two motions. The time interval between frames must be small to ensure that any acceleration of the moving components is negligible. We demonstrate that the algorithm provides precise motion estimates for a set of elementary two-motion configurations, including transparent pattern motion and motion boundaries, and show that it is robust in the presence of noise.

The two-motion algorithm we describe should be regarded as a basic component of a larger motion analysis system. It provides a more flexible method for estimating motion within local image regions. Other system components are required to select the local regions in which analysis is to be performed and to assemble results into an overall interpretation of scene motion.

II. Elementary Motion Configurations

As we have observed, the estimation of motion at a point in an image must be based on pattern information in a neighborhood of that point. We will refer to this neighborhood as the *motion analysis region.*

The size of the motion analysis region is a critical factor in motion estimation. It is important that it be small so that motion within the region can be described by a simple model. However, the region cannot be too small, or it may not encompass sufficient pattern detail to permit reliable motion estimation. The appropriate size is dictated by such factors as the size and velocity of objects in the scene.

These observations lead to two questions: How can we determine the "optimal" size for the analysis region, and what motion configurations may be expected to occur within regions that have this appropriately selected size? The answer to the first question is beyond the scope of the present paper, except to note that "foveation" [5], or split-and-merge procedures [19], might be used to control region size.

In answer to the second question, we have assembled a small set of *elementary motion configurations*, as shown in Fig. 1. The most common configuration is undoubtedly a single pattern undergoing coherent motion (Fig. 1(a)), but there are a number of commonly occurring configurations involving two motion components (Fig. 1(b)-(f)). More than two components can also occur, but these are relatively rare. Existing motion algorithms typically can deal with only one or two of these configurations adequately. Our objective in formulating a new motion analysis algorithm is that it should estimate correct motions in all of the configurations shown in this figure.

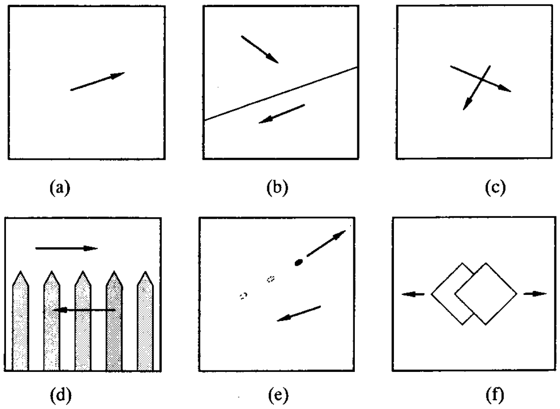


Fig. 1. Elementary local motion configurations: (a) Uniform motion of a single surface; (b) motion boundary; (c) transparent surfaces in motion; (d) "picket fence" motion; (e) masking; (f) two-component aperture effect.

The elementary local motion configurations are the following:

1. *Single Surface*: The analysis region contains a single pattern undergoing coherent (e.g., affine) motion.
2. *Motion Boundary*: The region contains two differently moving patterns separated by a distinct boundary.
3. *Transparent Surfaces*: The region contains two differently moving image patterns that appear superimposed. Examples include moving shadows, spotlights, reflections in a pond; etc., as well as actual transparent objects.
4. *"Picket Fence"*: The region contains small or thin foreground objects that move in front of a differently moving background, or the background appears through small gaps in the foreground. Foreground and background move coherently as two groups, although they may be disconnected in the image.
5. *Masking*: The region contains a dominant moving pattern and a second pattern that has low contrast or is small. The dominant pattern may mask the second in the elementary motion computation. An example is a football partially tracked by the camera in a sports broadcast.
6. *Two-Component Aperture Effect*: The aperture effect may be overcome by making the analysis region sufficiently large to include an entire object, but then, it is likely to contain two differently moving objects. In addition, features formed by the superposition of object patterns, such as T junctions in this example, may appear to move differently from either object.

This set of elementary motion configurations is intended to encompass the important cases in which two differently moving patterns occur within an image region. There may be other configurations of which we are not aware. The algorithm we propose in the next section can handle each of these and other configurations in which the image can be modeled as a combination of two coherently moving patterns.

III. Models for Local Motion

Motion estimation is based on an assumed model relating motion to observed image intensities. The traditional model used in optical flow computation postulates a single pattern moving uniformly within any local analysis region. We introduce a new model that postulates two such components.

A. Standard Single-Component Model

Let $I(x, y, t)$ be the observed gray-scale image at time t. Let R be the analysis region in which we wish to estimate motion.

The traditional model used in optical flow analysis [2], [10], [14] assumes that within the region R, the image may be represented as a pattern $P(x, y)$ moving with instantaneous velocity $\boldsymbol{p}(x, y)$. This motion field can be represented by velocity components in x and y: $\boldsymbol{p}(x, y) = (p_x(x, y), p_y(x, y))$. It is frequently assumed that this motion field is constant within R; the pattern P undergoes a simple rigid translation. More generally, the motion may be assumed to conform to other smoothly varying coherent motions, such as an affine transformation, that can be described with a small number of parameters. The analysis then seeks to estimate best values for these parameters. Formally

$$I(x, y, 0) = P(x, y),$$
$$I(x, y, 1) = P(x - p_x, y - p_y) = P^{\mathbf{p}}$$

and

$$I(x, y, t) = P(x - tp_x, y - tp_y) = P^{t\mathbf{p}} \quad (1)$$

where $P^{t\mathbf{p}}$ denotes the pattern P transformed by the motion $t\boldsymbol{p}$ (see Fig. 2(a)). Here, t is assumed to be a small time interval so that acceleration can be neglected. This model can represent only the first of the elementary motion configurations in Fig. 1 because it assumes that locally, there is only one coherent motion.

B. Proposed Two-Component Model

We introduce an alternative model for local motion, as shown in Fig. 2(b). This is based on the same assumption of locally coherent motion as in the standard model, but we now allow two motion components. Within the analysis region, the image is assumed to be a combination of two distinct image patterns P and Q, which have independent motions of $\boldsymbol{p}$ and $\boldsymbol{q}$:

$$I(x, y, 0) = P(x, y) \oplus Q(x, y)$$

and

$$I(x, y, t) = P^{t\mathbf{p}} \oplus Q^{t\mathbf{q}}. \quad (2)$$

Here, the $\oplus$ symbol represents an operator such as addition or multiplication that combines the two patterns.

The proposed two-motion model can represent (at least approximately) all of the elementary motion configurations in Fig. 1. For example, a motion boundary (Fig. 1(b)) can be represented as the sum of two patterns that are defined over the entire analysis region but that have zero amplitude over complementary portions of the region. If P moves to the right and its lower half is uniformly zero, whereas Q moves to the left and its upper half is uniformly zero, then the sequence $I(x, y, t)$ generated from their sum represents a scene whose upper half moves right and lower half moves left. Transparent motion of a reflection in a shop window also corresponds to the case in which $\oplus$ is addition, whereas patterns of light or shadow moving over a surface correspond to the case in which $\oplus$ is multiplication (Fig. 1(c)).

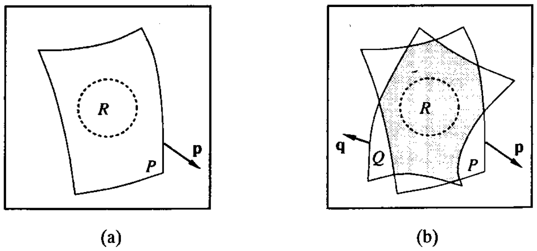


Fig. 2. Two models for local motion: (a) Traditional model with single motion: pattern P moves with velocity $\boldsymbol{p}$ within the analysis region R; (b) two-motion model: two patterns P and Q move with velocities $\boldsymbol{p}$ and $\boldsymbol{q}$.

IV. Estimating a Single Motion

We now review an algorithm for estimating a single image motion in accordance with the model of (1). In the next section, we show that this procedure for estimating single-component motion can be applied repeatedly to extract two motion components.

The single motion algorithm combines several techniques to achieve speed and precision. Although, individually, these techniques are not new, they are reviewed briefly here for completeness. First, we describe a basic *incremental-motion estimator* that can obtain estimates for motion, given that frame-to-frame displacements are small. Second, the precision of the estimator is enhanced through a successive *alignment* procedure. Finally, the range of the estimator is extended by implementing *coarse-fine alignment* within a pyramid structure.

A. Incremental-Motion Estimator

The problem of estimating the motion of an image region can be complicated and computationally expensive. However, if we restrict our consideration to small motions, it has been shown that there exists a simple, closed-form estimate [16], [17]. We review one derivation of this type of motion estimate. From (1), $I(x, y, t)$ can be expressed in terms of $I(x, y, t-1)$:

$$I(x, y, t) = I(x - p_x, y - p_y, t - 1). \quad (3)$$

(To simplify notation, let the frame interval be one unit of time.)

Adopting the standard "least squared error" approach, we wish to find the motion field $\boldsymbol{p} = (p_x, p_y)$ that minimizes the squared error:

$$Err' = \sum_{x,y \in R} (I(x,y,t) - I(x-p_x, y-p_y, t-1))^2. \quad (4)$$

Under the assumption that the displacement is small, the above equation can be simplified through truncated Taylor series expansion of $I(x,y,t)$:

$$\begin{aligned} I(x-p_x, y-p_y, t-1) \approx & I(x,y,t) - p_x I_x(x,y,t) \\ & - p_y I_y(x,y,t) - I_t(x,y,t) \quad (5) \end{aligned}$$

where

$$I_x = \frac{\partial I(x,y,t)}{\partial x}$$
$$I_y = \frac{\partial I(x,y,t)}{\partial y}$$
$$I_t = \frac{\partial I(x,y,t)}{\partial t}.$$

Then

$$Err = \sum_{x,y \in R} (I_t + p_x I_x + p_y I_y)^2. \quad (6)$$

The image motion is now obtained by setting the derivatives of (6) with respect to each of the parameters of the velocity components to zero and solving the resulting system of equations [12].

If the motion is modeled by simple translation, that is, $\boldsymbol{p} = (a_x, a_y)$, where a_x and a_y are constants, then the familiar optical flow equations [12], [13], [15] are obtained:

$$\begin{aligned} \left[\sum {I_x}^2\right] a_x + \left[\sum I_x I_y\right] a_y &= -\sum I_x I_t \\ \left[\sum I_x I_y\right] a_x + \left[\sum {I_y}^2\right] a_y &= -\sum I_y I_t. \quad (7) \end{aligned}$$

If, instead, we model motion as an affine transformation, $\boldsymbol{p}$ has six parameters:

$$p_x(x,y) = a_x + b_x x + c_x y$$
$$p_y(x,y) = a_y + b_y x + c_y y.$$

If the error in (6) is differentiated with respect to each of these parameters, a system of six equations with six unknowns is obtained, and it is shown at the bottom of this page.

This system is solved for the coefficients of the affine transformation.

B. Alignment

The above estimation method is accurate, in general, only when the frame-to-frame displacements due to motion are a fraction of a pixel so that the Taylor series approximation is meaningful. The precision of the estimates can be significantly improved through an iterative alignment procedure [3], [5]. After an initial estimate of motion is obtained, the first image is shifted toward the second to compensate for the estimated displacement. The motion estimation procedure is then repeated between the shifted first image and the original second image to obtain an estimate of any residual velocity. These shift and estimate steps are iterated to bring the first image into alignment with the second, thereby progressively reducing the frame-to-frame displacement and creating conditions in which the incremental-motion estimator is most accurate.

Let $\boldsymbol{p}_k$ be the velocity estimate obtained after the kth iteration of the alignment process. Let $\boldsymbol{p}_0$ be the *a priori* estimate of velocity before analysis begins. Typically, we assume $\boldsymbol{p}_0 = 0$. Steps of the alignment procedure during the kth iterations ($k \geq 1$) are as follows (see Fig. 3):

1. The first image $I(x,y,t-1)$ is shifted or *warped* toward the second image $I(x,y,t)$ in accordance with the velocity estimated $\boldsymbol{p}_{k-1}$ obtained on the previous iteration:

$$I_{k-1} = I(x - p_{x\,k-1}, y - p_{y\,k-1}, t-1).$$

2. The incremental-motion estimator is applied to $I^{\boldsymbol{p}_{k-1}}(x,y,t-1)$ and $I(x,y,t)$ to obtain an estimate $\Delta\boldsymbol{p}_k$ of residual motion.
3. The estimated motion is updated:

$$\boldsymbol{p}_k = \boldsymbol{p}_{k-1} + \Delta\boldsymbol{p}_k.$$

When initial displacements are within range of the incremental motion estimator, this alignment procedure generally converges rapidly, usually achieving its limiting accuracy within two or three iterations.

C. Coarse-Fine Alignment

The range of the motion estimation process can be extended to the general case of large displacements by implementing alignment within a multiresolution (pyramid) structure (Fig. 4).

A Gaussian pyramid is constructed for each of the source image frames $I(x,y,t-1)$ and $I(x,y,t)$. This pyramid is a sequence of copies of the original image in which both resolution and sample density are reduced by powers of 2. Let $G_{t,\ell}$ be the ℓth pyramid level for image $I(x,y,t)$. The zero

$$\begin{bmatrix} \sum {I_x}^2 & \sum x{I_x}^2 & \sum y{I_x}^2 & \sum I_x I_y & \sum xI_x I_y & \sum yI_x I_y \\ \sum x{I_x}^2 & \sum x^2{I_x}^2 & \sum xy{I_x}^2 & \sum xI_x I_y & \sum x^2 I_x I_y & \sum xyI_x I_y \\ \sum y{I_x}^2 & \sum xy{I_x}^2 & \sum y^2{I_x}^2 & \sum yI_x I_y & \sum xyI_x I_y & \sum y^2 I_x I_y \\ \sum I_x I_y & \sum xI_x I_y & \sum yI_x I_y & \sum {I_y}^2 & \sum x{I_y}^2 & \sum y{I_y}^2 \\ \sum xI_x I_y & \sum x^2 I_x I_y & \sum xyI_x I_y & \sum x{I_y}^2 & \sum x^2{I_y}^2 & \sum xy{I_y}^2 \\ \sum yI_x I_y & \sum xyI_x I_y & \sum y^2 I_x I_y & \sum y{I_y}^2 & \sum xy{I_y}^2 & \sum y^2{I_y}^2 \end{bmatrix} \begin{bmatrix} a_x \\ b_x \\ c_x \\ a_y \\ b_y \\ c_y \end{bmatrix} = -\begin{bmatrix} \sum I_x I_t \\ \sum xI_x I_t \\ \sum yI_x I_t \\ \sum I_y I_t \\ \sum xI_y I_t \\ \sum yI_y I_t \end{bmatrix}$$

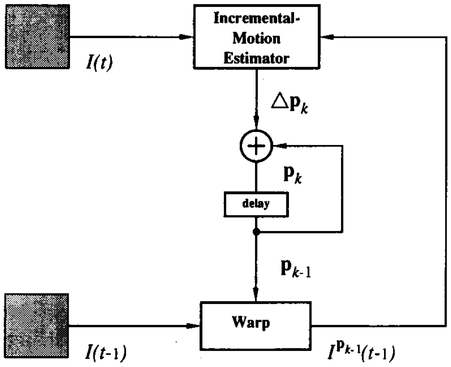


Fig. 3. Precise motion estimates obtained through feedback and successive alignment.

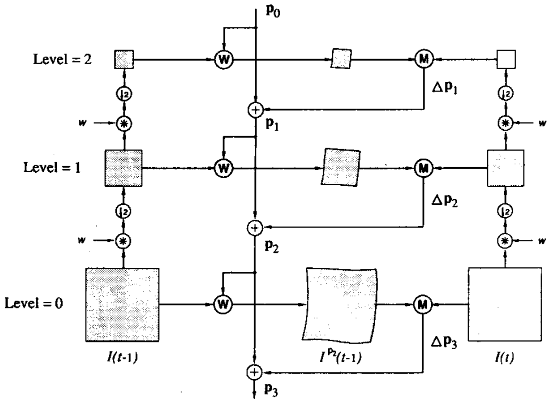


Fig. 4. Diagram showing the sequence of operations in pyramid-based coarse-fine alignment.

level is identical to the source image, i.e., $G_{t,0} = I(x,y,t)$; the ℓth level is obtained by convolving the $\ell - 1$ level with a small kernel filter w followed by subsampling [4]:

$$G_{t,\ell} = [G_{t,\ell-1} * w]_{\downarrow 2}.$$

Here, $\downarrow 2$ indicates that the quantity in brackets has been subsampled by 2 in both x and y; every other row and column are discarded.

Motion analysis begins at a low resolution level of the image pyramid. The sample distance at level ℓ is 2^ℓ times that of the original image. This means correspondingly larger image velocities can be estimated. At each successive iteration, the shift and estimate steps are performed on the next higher resolution pyramid level. Thus, if level ℓ is processed at iteration k, then the shift (or warp) estimated at level $\ell + 1$ is applied to pyramid level $G_{t-1,\ell}$ to form $G^{\boldsymbol{p}_{k-1}}_{t-1,\ell}$, and the residual $\Delta \boldsymbol{p}_k$ is computed between this and the corresponding level of the second pyramid $G_{t,\ell}$. Shifting ensures that residual displacements remain less than a sample distance as the procedure moves to each higher resolution pyramid level until full resolution is reached. Thus, coarse-fine tracking can efficiently estimate velocities of many pixels per frame time at accuracies of a small fraction of a pixel [2], [3], [6], [10]. Note that this process can be represented in terms of the loop in Fig. 3, with the addition of a control process that decreases the scale of analysis at each cycle of the loop.

V. Estimating Two Motions

We now consider the analysis of motion described by the two-component model (2). If a direct extension of the least squares estimation technique is attempted, it becomes necessary to first estimate spatial and temporal derivatives of both moving patterns P and Q. However, these derivatives can only be estimated if the patterns are separated prior to motion analysis, i.e., the image is segmented.

Alternative approaches that simultaneously estimate two-component motion without segmentation have been proposed. Examples include the use of Hough transform techniques, cross correlation, and "direct" estimation [7], [9], [21]. However, these are computationally difficult and may not provide results of the desired precision. The present approach obviates both the need for segmentation and the need to estimate two motion components simultaneously.

The key observation for the present approach is that if one of the motion components and the combination rule $\oplus$ are known, it is possible to remove that component pattern from the images and compute the other motion using the single-component motion algorithm without determining patterns P or Q themselves. In that which follows, we will assume that the combination operation is addition. The case of multiplication can also be turned into addition by taking the logarithm of the images.

Suppose, for the moment, that motion $\boldsymbol{p}$ is known so that only motion $\boldsymbol{q}$ must be determined. The pattern component P moving at velocity $\boldsymbol{p}$ can be removed from the image sequence by shifting each image frame by $\boldsymbol{p}$ and subtracting it from the following frame. The resulting sequence will contain only patterns moving with velocity $\boldsymbol{q}$.

Let D_1 and D_2 be the first two frames of this difference sequence, which were obtained from three original frames. From (2)

$$\begin{aligned} D_1 &\equiv I(x,y,2) - I^{\boldsymbol{p}}(x,y,1) \\ &= (P^{2\boldsymbol{p}} + Q^{2\boldsymbol{q}}) - (P^{2\boldsymbol{p}} + Q^{\boldsymbol{q}+\boldsymbol{p}}) \\ &= Q^{2\boldsymbol{q}} - Q^{\boldsymbol{q}+\boldsymbol{p}} \\ &= (Q^{\boldsymbol{q}} - Q^{\boldsymbol{p}})^{\boldsymbol{q}}, \end{aligned}$$

$$\begin{aligned} D_2 &\equiv I(x,y,3) - I^{\boldsymbol{p}}(x,y,2) \\ &= (P^{3\boldsymbol{p}} + Q^{3\boldsymbol{q}}) - (P^{3\boldsymbol{p}} + Q^{2\boldsymbol{q}+\boldsymbol{p}}) \\ &= Q^{3\boldsymbol{q}} - Q^{2\boldsymbol{q}+\boldsymbol{p}} \\ &= (Q^{\boldsymbol{q}} - Q^{\boldsymbol{p}})^{2\boldsymbol{q}}. \end{aligned} \tag{8}$$

The sequence $\{D_n\}$ now consists of a new pattern $Q^{\boldsymbol{q}} - Q^{\boldsymbol{p}}$ moving with a single motion $\boldsymbol{q}$, that is, $D_n = (Q^{\boldsymbol{q}} - Q^{\boldsymbol{p}})^{n\boldsymbol{q}}$. Thus, the motion $\boldsymbol{q}$ can be computed from the two difference images D_1 and D_2 using the single-motion estimation technique described in the previous section.

In an analogous fashion, the motion $\boldsymbol{p}$ can be recovered when $\boldsymbol{q}$ is known. The observed images $I(x,y,t)$ are shifted by $\boldsymbol{q}$, and a new difference sequence is formed:

$$D_n{}' = I(x, y, n+1) - I^{\boldsymbol{q}}(x, y, n).$$

This sequence is the pattern $P^{\boldsymbol{p}} - P^{\boldsymbol{q}}$ moving with velocity $\boldsymbol{p}$: $D_n{}' = (P^{\boldsymbol{p}} - P^{\boldsymbol{q}})^{n\boldsymbol{p}}$. Therefore, $\boldsymbol{p}$ can be recovered using the single-motion estimation algorithm.

Note that the shift and subtract procedure removes, or "nulls," one moving pattern from the image sequence without determining what that pattern is and without explicit segmentation. If the combination rule in (2) is multiplication, then a shift-and-divide procedure in (8) would achieve the same nulling function, yielding $D_n = (Q^{\boldsymbol{q}}/Q^{\boldsymbol{p}})^{n\boldsymbol{q}}$ when $\boldsymbol{p}$ is known.

In practice, of course, neither motion $\boldsymbol{p}$ nor $\boldsymbol{q}$ is known *a priori*. However, it is possible to recover both motions precisely if we start with even a very crude estimate of either. It is generally sufficient to assume $\boldsymbol{p} = 0$ in order to obtain a first estimate of $\boldsymbol{q}$ if no better *a priori* information is available.

Two-component motion analysis can therefore be formulated as an alternating iterative refinement procedure (Fig. 5). Let $\boldsymbol{p}_n$ and $\boldsymbol{q}_n$ be the estimates of motion after the nth cycle. Estimates alternate between $\boldsymbol{p}$ and $\boldsymbol{q}$; therefore, if $\boldsymbol{p}$ is obtained on even-numbered cycles, $\boldsymbol{q}$ is obtained on odd cycles. Steps of the procedure are as follows:

1. Set an initial estimate for the motion $\boldsymbol{p}_0$ of pattern P.
2. Form the difference images D_1 and D_2 as in (8) using the latest estimate of $\boldsymbol{p}_n$.
3. Apply the coarse-fine single-motion estimator to D_1 and D_2 to obtain an estimate of $\boldsymbol{q}_{n+1}$.
4. Form new difference images D_1 and D_2 using the estimate $\boldsymbol{q}_{n+1}$.
5. Apply the single-motion estimator to the new sequence D_1 and D_2 to obtain an update $\boldsymbol{p}_{n+2}$.
6. If a desired level of precision (stability) has been attained, then stop; else, repeat starting at Step 2.

In the cases we have tried, convergence of this process is fast; with artificially generated image sequences, the correct transformations are recovered to within roughly 1% after three to five cycles, regardless of the initial guess of $\boldsymbol{p}_0$. We have not attempted to determine analytically the conditions under which the algorithm is guaranteed to converge.

When this two-motion algorithm is applied to a region containing only one moving pattern, it will detect that motion on the first iteration but will pick up "motion" of noise in the second. In practice, a test will be required to detect this situation. One way of detecting such a situation is to use the estimated motion to register the difference images used in the computation and compare the mean square of the registered images with the mean square error of the unregistered difference images.

VI. Examples of Two-Motion Analysis

We have tested the two-motion algorithm with several examples of the elementary motion configurations shown in Fig. 1. We have used both artificial sequences constructed from moving random noise patterns and real images of complex natural scenes. In all examples in this section, the analysis region R was taken to be the entire image, and the images were 256 × 256 or 256 × 200 pixels in size. In all cases, coarse-fine computations began at pyramid level three and moved to level zero. The initial motion estimate for both components was taken to be zero. When artificial sequences were used, the actual velocities were known, and the accuracy of the estimate could be determined. All computations were performed on a Sun SparcStation 1. Each full iteration of the algorithm described in the previous section required roughly 10 s.

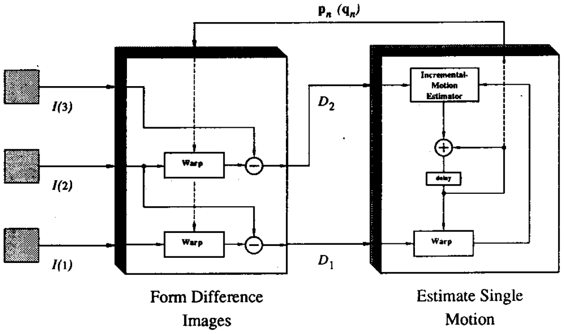


Fig. 5. Two-stage computation for recovering two motion components from three input images. The loop inside the right-hand box represents the coarse-fine iteration detailed in Fig. 4. The outer loop increments each time the inner loop is completed.

A. Example 1: Transparent Motion

A synthetic image sequence showing transparent motion was constructed by adding two random dot patterns P and Q, where one translated (8,0) pixels between successive frames, and the other translated (0,8) pixels. The appearance of this sequence is of one transparent textured surface sliding over a second opaque surface. The two correct translational components of the original sequences were recovered after three cycles of the coarse-fine process. Other choices of initial guess produced similar results.[1] Actual recovered translations were (8.04, 0.01) and (0.01, 8.03). Unfortunately, the results of this example cannot easily be displayed in a still image. In a video sequence showing the compensated difference images D_n, it is easily seen that the two motions have been accurately separated.

A second example involving additive transparency is shown in Fig. 6. In this case, a sequence was captured with a moving video camera showing a face reflected in the glass covering a print of Escher's "Three Worlds." A single frame from this sequence is shown in Fig. 6(a). As the camera moved, the image reflected in the glass and the image in the print moved differently. These two motions were computed from this sequence and used to produce the compensated difference images (frames from D_n) shown in Fig. 6(b) and (c). In Fig. 6(b), the reflected image (which is barely visible in Fig. 6(a))

[1] The importance of the initial motion estimates has not been studied systematically. Clearly, convergence can only be obtained if the error in the initial estimates falls within the range of velocities that can be detected by the single motion algorithm. This range is large because the algorithm is implemented within a pyramid structure.

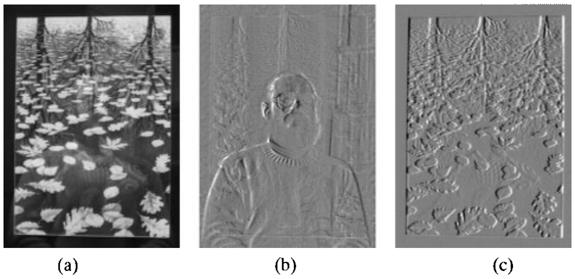


Fig. 6. Transparent motion. A sequence of images was obtained as the camera moved, showing a face reflected in the glass of a framed picture: (a) One frame from the sequence; (b) difference of two consecutive frames after registration using the computed motion of the picture. The picture cancels out, and the face structure is visible; (c) difference of two consecutive frames after registration, using the computed motion of the reflected face. The face is canceled out, and only the picture structure is visible.

is revealed, showing that the other component was registered accurately. In Fig. 6(c), the reflected image has been nulled.

B. Example 2: Motion Boundary

The second example demonstrates motion estimation at a boundary. This sequence was constructed from two random noise fields that are not transparent but are forming foreground and background regions. The upper left field is in the background, moving with velocity $(6.831, 2.331)$. The foreground field moving with velocity of $(-3.863, 1.024)$ covers a region in the lower half of the picture. These displacements correspond to a motion parallel to the boundary for the foreground segment and a velocity oblique to the boundary for the background. In this case, the sequence is not precisely the sum of two uniformly moving patterns because a small area of the background is hidden, or occluded, by the foreground object on each frame. In spite of this minor violation of the sequence structure assumed in the two-component motion model, the algorithm successfully recovers the motion components. The translation components determined by the algorithm after two iterations are $(6.828, 2.322)$ and $(-3.845, 1.041)$. The result of compensating for one of the estimated displacements and subtracting successive frames is displayed Fig. 7(c). It can be seen that the estimated displacement corresponds very accurately to the motion in one of the two regions, resulting in that region being blank in the compensated difference image. In this example, knowledge of the two motions leads directly to an accurate segmentation of the image. For comparison, an optical flow computation [3] results in the compensated difference image in Fig. 7(d). Here, the pattern is canceled over most of the image area, indicating accurate motion compensation, but does not cancel near the boundary.

C. Example 3: Masking

A second sequence of real images was digitized to demonstrate motion recovery when one motion pattern predominates, and "masks," the second pattern as in Fig. 1(e). This sequence is an "aerial photograph": a small toy tank moves rapidly in front of a large moving background of toy roads and trees. One frame of this sequence is shown in Fig. 8(a). Because the motion of the foreground object is roughly equal to its own size, it would be difficult to select a window within which this motion would dominate. However, the two-motion algorithm obtains accurate estimates of both background and foreground motions. The background cancelation is shown in Fig. 8(b) and the foreground cancelation in Fig. 8(c). Note the absence of the moving vehicle in this last image. Accurate estimation of both motions is obtained in spite of the fact that the combination of foreground and background components is not strictly additive.

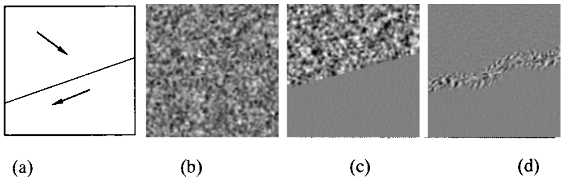


Fig. 7. Motion boundary: (a) Sequence of frames was constructed in which regions of random texture moved; (b) one image in the sequence; (b) multiple motion algorithm used to recover both motions; (c) when one image is shifted by one of these motions and a difference image is formed, the corresponding moving pattern is canceled, and the boundary is revealed; (d) if an optical flow algorithm is applied instead, erroneous motion estimates are obtained along the boundary, as is apparent when the estimated motion is used to register successive frames, and a difference is formed.

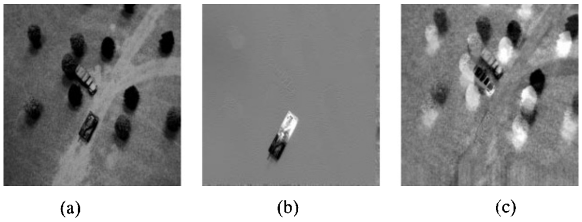


Fig. 8. Masking. A small moving object may be obscured when viewed against a larger, differently moving background: (a) One frame from the sequence; (b) difference of two consecutive frames after registration using the background motion. The background is canceled out, and the tank is visible; (c) difference of two consecutive frames after registration using the tank motion. The tank is canceled out, and only the background structure is visible.

D. Example 4: Two-Component Aperture Effect

An example involving both transparency and a two-component aperture effect is shown in Fig. 9. The image sequence in this case consists of the sum of two uniform squares moving diagonally in opposite directions, as in Fig. 1(f). In this case, the actual motions were $(2.0, 2.0)$ and $(-2.0, -2.0)$. An optical flow computation [3] results in the flow field shown in Fig. 9(c). Note that almost all flow vectors point in directions other than the direction of actual motion. Some vectors correspond to the well-known aperture effect, and others correspond to the apparent motion of features formed by the superposition of two differently moving patterns. Clearly, it would be very difficult to recover accurate estimates of object motions from such a flow field. However, when the two-component motion algorithm is applied, actual object motions are recovered to machine precision after only two iterations.

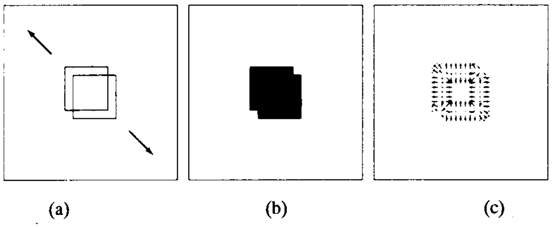


Fig. 9. Two component aperture effects: (a) Input configuration; (b) one frame from sequence; (c) optical flow field computed from two frames of the sequence. Note that the complex pattern of flow does not correspond to the motion of either object. When the two-component algorithm is applied, both motions are accurately recovered.

E. Example 5: "Picket Fence"

The final example (Fig. 10) shows an image sequence in which a crowd of people is viewed through a complex pattern of tree branches. The camera is translating and rotating; therefore, the foreground trees and background crowd are seen to move differently. Because the motions include dilation and rotation as well as translation, we must estimate two affine transformations. This is an example of a "picket fence" configuration (Fig. 1(d)). In spite of many violations of the additivity assumption due to occlusion and exposure, convergence is reached after four iterations. In order to demonstrate the accuracy of the foreground and background motion estimates, we have generated two "temporal average" images after registering the three input images using the two estimated motions (Fig. 10(c) and (d)). In each of these, the registered areas are sharp, whereas the rest of the image is blurred due to the image motion. For reference, an unregistered temporal average is shown in Fig. 10(b).

VII. Quantitative Experiments

A. Stability Analyses

The examples shown in the preceding section suggest that the algorithm that we have described is surprisingly robust with respect to violations of the assumptions about image sequence structure expressed in (2). Of the examples shown, only Example 1, which involves transparency, can be exactly represented as the sum of two coherently moving patterns. In the others, some areas appear or disappear from frame to frame. In the case of the tree scene (Example 5), there are also objects within the analysis region that move with velocities unrelated to either of the two major coherent components. Nevertheless, the registration of the major components is fairly accurate. In the case of the synthetic images where the motions are known exactly, these values are recovered precisely in spite of violations of assumptions.

1) Experiments: Two experiments were performed to determine the limits of the algorithm's performance when applied to image sequences that do not precisely conform to the two-component motion model. In both cases, the test sequence was the sum of unfiltered Gaussian noise images with standard deviation equal to 15 gray levels. Each component moved with a speed of 3 pixels/frame, with one to the right and the other to the left.

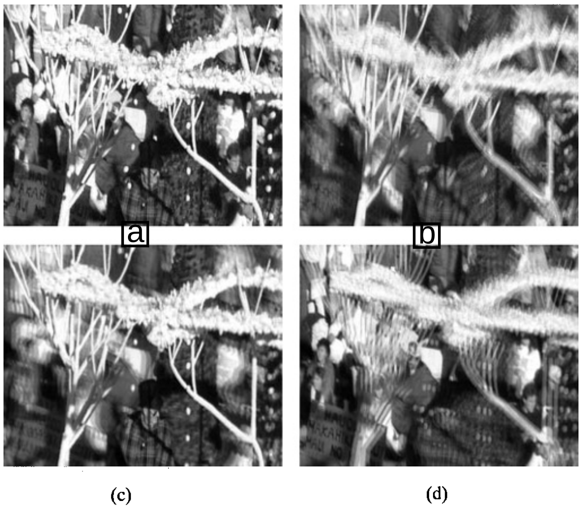
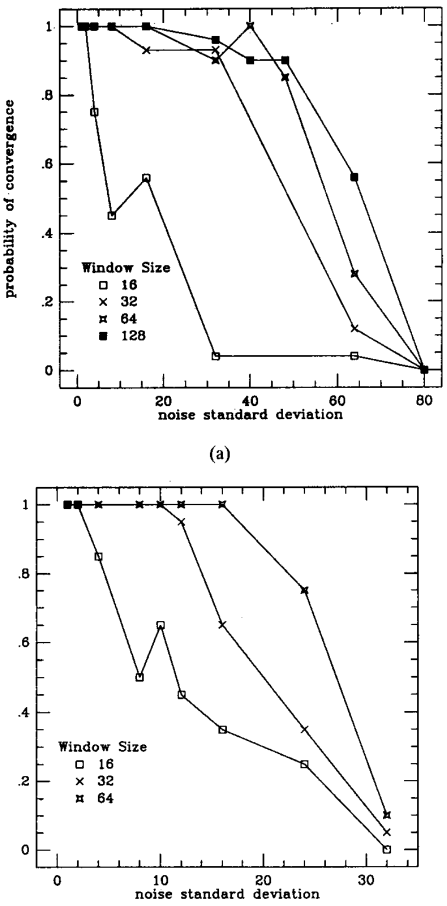


Fig. 10. "Picket fence." A crowd is viewed through the branches of a foreground tree. The camera is moving so that the foreground and background appear to move in different directions: (a) One frame from the original sequence; (b) averaging three consecutive frames from the original sequence (no motion compensation). The entire scene is blurred; (c) averaging three frames after registration with the foreground motion. The trees are sharp, whereas the background is blurred; (d) averaging three frames after registration with the background motion. The background remains sharp, whereas the foreground is blurred.

In the first experiment, temporally uncorrelated noise was added to the motion sequence. This simulates the effect of image occlusion since regions of the image that appear or disappear from frame to frame produce local changes in intensity that are uncorrelated in time. In the second experiment, a moving uniformly distributed noise pattern was added to the original two-component sequence. This simulates the effect of motions that do not fit the model of either coherent motion being estimated. Note that noise signals are distributed uniformly over the analysis region in these experiments, although the conditions that these experiments are designed to simulate (such as occlusion effects) are generally localized in natural images. This difference is not critical, however, since the contributions are summed over the analysis region.

In each experiment, two factors were varied: the amplitude of the interfering signal and the size of the analysis region. Two characteristics of algorithm performance were measured: the likelihood that the algorithm successfully isolated the two motion components after 20 cycles of the algorithm (ten for each motion component) and the average RMS error in those estimates with respect to the true velocities. The region size was varied over a wide range because increased size may be expected to decrease sensitivity of the algorithm to noise. In both experiments, only uniform displacement was estimated, rather than a more complex transformation.

2) Results: Fig. 11(a) shows the results using uncorrelated noise. The standard deviation of the noise is on the abscissa. Since the noise was uniformly distributed, the range of the noise is the standard deviation multiplied by 1.732. The probability that the two-motion algorithm converged to within 20% of the correct velocities within ten cycles of the "estimate-



(a)

(b)

Fig. 11. Probability of convergence as a function of noise level. The abscissa shows noise standard deviation. The ordinate shows probability of convergence to within 20% of the correct motion estimates within ten iterations. The error is defined as the rms error divided by the rms amplitude of the velocities, and thus, convergence requires that both motions be reasonably well estimated. The various curves correspond to window sizes ranging from 16 x 16 to 128 x 128 for the uncorrelated noise and 16 x 16 to 64 x 64 for the moving noise: (a) Uncorrelated noise: new samples of noise were generated for each frame; (b) moving noise: one sample of noise was generated and then moved upwards by three pixels on each frame.

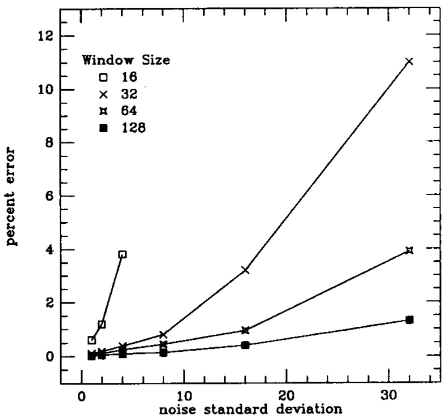


Fig. 12. Percentage RMS error when probability of convergence is above 50%. The abscissa shows the noise standard deviation. Curves show window sizes ranging from 16 x 16 to 128 x 128.

subtract" analysis process is shown on the ordinate. Each probability estimate is based on 30 trials with the same signal but independent samples of noise. Four curves are shown, representing window sizes of 16 x 16, 32 x 32, 64 x 64, and 128 x 128 pixels.

A number of characteristics are worthy of note. First, with little or no noise, even a window size of only 16 x 16 is sufficient for reliable convergence of the algorithm. However, for this smallest window size, the results are sensitive to noise, and by a noise standard deviation of about three gray levels, the process is already rather unreliable. This is a relatively high noise value, corresponding to a signal-to-noise ratio of 5 since the individual "signal" components have a standard deviation of only 15 gray levels. For larger window sizes, however, the process is very resistant to the effects of uncorrelated noise. It is not until the signal-to-noise ratio falls well below 1 that the probability of convergence drops below 90%. Furthermore, for these stimuli, at least, there is only a slight benefit in increasing the window size above 32 x 32.

The results of the second experiment are shown in Fig. 11(b). A third motion component is introduced, and it moves at the same speed as the original two (3 pixels per frame) but upward rather than right or left. Again, the abscissa shows the noise component standard deviation (note the difference in scale), and the ordinate shows the probability of convergence within 20% of the correct signal velocities. For the 16 x 16 window size, the results are very similar to those for the uncorrelated noise; the algorithm is rather noise sensitive. For the larger window sizes, performance is reliable down to a signal-to-noise ratio of about 2. Beyond this level, performance decays rapidly. This is not surprising since in these stimuli, the signal components and the noise are almost identical. When the noise component approaches the signal components in amplitude, the algorithm begins to track the noise instead of one of the signal components. Thus, there is no possibility of correctly estimating the signal velocities when the signal-to-noise ratio is less than 1. However, it is clear that for moderate levels of extraneous motion, the algorithm continues to provide meaningful estimates.

An additional measure of the robustness of this algorithm is shown in Fig. 12, which shows the RMS deviation of the estimated velocities from the true values for the cases in which convergence was obtained. Clearly, this is only of interest when the probability of convergence is high and when the estimated variation is considerably smaller than the criterion for convergence. The figure shows values as a function of uncorrelated noise levels for the four window sizes. For all but the smallest window size, the expected error grows gradually and smoothly with noise level. Performance overall is highly accurate. Similar precision is found in the case of the moving noise when conditions yielding similar probabilities of convergence and window sizes are compared.

3) Conclusions: These results suggest that the performance of the algorithm is robust, at least with respect to the violations of assumptions introduced here. This is of considerable importance since in real image sequences, the assumptions of the two-motion model will never be satisfied precisely. These experimental results help explain the good performance of the algorithm on several of the examples shown in the previous section, particularly those involving real images.

VIII. Summary and Comments

Most current approaches to motion analysis are based on a single motion assumption; when an image sequence is viewed through a sufficiently small analysis window over a sufficiently short interval of time, it may be modeled as a single pattern undergoing uniform motion. This assumption holds and can lead to accurate motion estimates within many local regions of a typical image sequence. It fails, however, when even a small analysis window contains two or more differently moving patterns, such as along the boundary between a moving object and its background and where semi-transparent surfaces or patterns of light move over other surfaces. Such failures lead to the incorrect interpretation of a scene.

Techniques to address limitations of the single motion model have been proposed, but these introduce other analysis problems. Image segmentation, for example, can be used to control the placement of local analysis regions to ensure that regions do not cross motion boundaries. However, this presents a "chicken-egg" dilemma since segmentation processes must often rely on motion analysis to detect such boundaries. In addition, conventional segmentation cannot handle transparency. Methods that simultaneously estimate two motions within a region may be limited in their ability to distinguish similar motions since each motion component constitutes noise in the signal as it is used to estimate the other component.

We propose an alternative approach to the analysis of multiple motions that largely overcomes limitations of previous methods. The components are estimated one at a time using a single motion algorithm. Once an initial estimate of one component has been obtained, the associated pattern is largely removed from the image sequence through a shift-and-subtract procedure. Three frames of the original sequence are used to prepare two difference frames that can be used to estimate the second motion, again using a single motion algorithm. These steps are then repeated to obtain a more accurate estimate of the first motion. A few iterations generally suffice to isolate motion components and obtain highly precise motion estimates. Speed, precision, and robustness are obtained by implementing all computations within a pyramid framework.

We show that the new approach to motion estimation can handle a variety of basic two-component motion configurations in a unified way. The same computation steps can obtain precise motion estimates at motion boundaries, identify motions of transparent patterns, and detect small or low contrast moving patterns in the presence of large, high-contrast patterns. The approach does not require explicit image segmentation to obtain precise estimates of each component motion.

Several issues that are important to full motion analysis have not been addressed in this paper and require further research. We assume that motion analysis is performed within local regions that have been selected to have, at most, two differently moving pattern components. This relaxes the single motion constraint imposed in most past approaches and means that the analysis regions can generally be much larger than is possible with conventional approaches. However, when more than two motions occur within a given region, it is then necessary to reposition and/or reduce the size of the region. We have not addressed the problems of how to detect whether more than two motions have occurred or how to automatically select new analysis regions. Again, an advantage of the present approach is that it does not require segmentation to obtain precise motion estimates of two pattern components. This should provide a powerful starting point for subsequent segmentation.

Finally, it should be noted that our approach assumes that both moving pattern components have constant velocity over the three frames used in analysis. This can be a significant restriction if objects are accelerating, and the frame rate is low.

References

[1] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-7, no. 4, pp. 384–401, July 1985.

[2] P. Anandan, "A unified perspective on computational techniques for the measurement of visual motion," in *Proc. Int. Conf Comput. Vision* (London), May 1987, pp. 219–230.

[3] J. R. Bergen and E. H. Adelson, "Hierarchical, computationally efficient motion estimation algorithm," *J. Opt. Soc. Amer. A.*, vol. 4, p. 35, 1987.

[4] P. J. Burt, "Fast filter transforms for image processing," *Comput. Graphics Image Processing*, vol. 16, pp. 20–51, 1981.

[5] P. J. Burt *et al.*, "Object tracking with a moving camera, An application of dynamic motion analysis," in *Proc. IEEE Workshop Visual Motion* (Irvine, CA), Mar. 1989, pp. 2–12.

[6] P. J. Burt, C. Yen, and X. Xu, "Multi-resolution flow-through motion analysis," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.* (Washington, DC), June 1983, pp. 246–252.

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

[8] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, Nov. 1984.

[9] B. Girod and D. Kuo, "Direct estimation of displacement histograms," in *Proc. Image Understanding Machine Vision Opt. Soc. Amer.* (Cape Cod), June 1989, pp. 73–76.

[10] D. J. Heeger, "Optical flow using spatiotemporal filters," *Int. J. Comput. Vision*, vol. 1, pp. 279–302, 1988.

[11] E. C. Hildreth, *The Measurement of Visual Motion*. Cambridge, MA: MIT Press, 1983.

[12] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.

[13] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185–203, 1981.

[14] B. K. P. Horn and E. J. Weldon, "Direct methods for recovering motion," *Int. J. Comput. Vision*, vol. 2, no. 1, pp. 51–76, June 1988.

[15] D. Keren, S. Peleg, and R. Brada, "Image sequence enhancement using sub-pixel displacements," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.* (Ann Arbor, MI), pp. 742–746, June 1988.

[16] J. O. Limb and J. A. Murphy, "Estimating the velocity of moving images in television signals," *Comput. Graphics Image Processing*, vol. 4, no. 4, pp. 311–327, Dec. 1975.

[17] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Image Understanding Workshop*, 1981, pp. 121–130.

[18] D. W. Murray and B. F. Buxton, "Scene segmentation from visual motion using global optimization," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 2, pp. 220–228, Mar. 1987.

[19] T. Pavlidis, *Structural Pattern Recognition*. New York: Springer, 1977.

[20] S. Peleg and H. Rom, "Motion based segmentation," in *Proc. Int. Conf. Patt. Recogn.* (Atlantic City, NJ), June 1990, pp. 109–113, vol. 1.

[21] M. Shizawa and K. Mase, "Simultaneous multiple optical flow estimation," in *Proc. Int. Conf. Patt. Recogn.* (Atlantic City, NJ), June 1990, pp. 274–278.

James R. Bergen received the A. B. degree in mathematics and psychology from the University of California at Berkeley in 1975 and the Ph.D. degree in biophysics and theoretical biology from the University of Chicago in 1981.

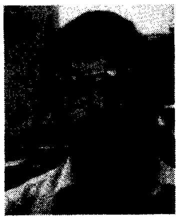
During 1981–1982, as a postdoctoral fellow at Bell Laboratories, Murray Hill, NJ, he studied the influence of image spatial structure on the rapid visual perception and the processing of visual texture. In 1982, he joined the David Sarnoff Research Center, Princeton, NJ, as a member of the Advanced Image Processing Research Group. Since 1991, he has been a Senior Member of Technical Staff. His research interests include both human and computer vision. His current active areas of interest include image processing (particularly motion and texture analysis), with applications to computer vision, image interpretation, and signal processing.

Peter J. Burt (M'80) received the B. A. degree in physics from Harvard University, Cambridge, MA, in 1968 and the Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 1976.

From 1968 to 1972, he conducted research in sonar, particularly in acoustic imaging devices, at the U. S. Navy Underwater Systems Center, New London, CT. As a postdoctural fellow, he has studied both natural and computer vision at New York University (1976–1978), Bell Laboratories (1978–1979), and the University of Maryland (1979–1980). He was a member of the engineering faculty at Rensselaer Polytechnic Institute, Troy, NY, from 1980 to 1983. In 1983, he joined the David Sarnoff Research Center, Princeton, NJ, and has been Head of the Advanced Image Processing Research Group since 1984. He is active in the development of fast algorithms and computing architectures for real-time computer vision. Applications include vehicle navigation and object recognition.

Rajesh Hingorani received the B. Tech. degree in electrical engineering from the Indian Institute of Technology, Kanput, India, in 1978 and the M. S. degree in computer systems engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1979.

From 1980 to 1982, he was a Member of the Technical Staff at the David Sarnoff Research Center, Princeton, NJ. He is currently a Member of the Technical Staff at the HDTV Systems Development Department, AT&T Bell Laboratories, Murray Hill, NJ. His research interests include image compression, motion analysis, digital signal and image processing, image understanding, and computer vision.



Shmuel Peleg (M'84) received the B.Sc. degree in mathematics from the Hebrew University of Jerusalem, Jerusalem, Israel, in 1976 and the M.Sc. and Ph.D. degrees from the University of Maryland, College Park, in 1978 and 1979, respectively.

He has been the Chairman of the Institute of Computer Science at the Hebrew University of Jerusalem since 1990, where he has been a faculty member since 1980. Over various periods from 1979 to 1992, he was a visiting researcher at the University of Maryland, New York University, and the David Sarnoff Research Center. His current research interests are in the computer vision area.