Computing Two Motions from Three Frames

James R. Bergen Peter J. Burt
Rajesh Hingorani Shmuel Peleg*

David Sarnoff Research Center
Subsidiary of SRI International
Princeton, NJ 08543-5300, USA

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 undergoing a simple translation: even complex motion will appear as a uniform displacement when viewed through a sufficiently small window. This assumption fails in a number of common situations. For example, transparent surfaces moving past one another yield two motion components at each point. More important, the assumption fails along the boundary between two differently moving image regions.

We propose an alternative formulation in which there may be two distinct patterns undergoing coherent motion within a given local analysis region. We then present an algorithm for the analysis of two-component motion. 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.

1 Introduction

The optical flow approach to motion analysis has been based on a *single-component model* of local image motion: even complex motion will be indistinguishable from a single pattern undergoing simple translation when viewed through a sufficiently small window, over a sufficiently short interval of time [8]. However, a single-motion model is inadequate for a number of important situations that commonly occur in image sequences. For example, transparent surfaces moving past one another yield two motion components at a point, as do patterns of light and shadow moving over a surface. 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 a scene.

This problem at boundaries is a consequence of the introduction of smoothness constraints in optical flow computation, whether this is done explicitly or not [7, 1]. In an effort to increase accuracy near boundaries, recent approaches allow a small number of discontinuities between smoothly varying regions. In this approach good quality motion analysis depends on image segmentation, while segmentation depends in turn on good quality motion information. Methods have been proposed that combine computation of motion and image segmentation, relying on successive refinement to converge to a stable interpretation of the scene [9, 10]. Examples of this approach include Markov Random Field models incorporating 'line processes' to decouple motion estimation across boundaries, and 'brittle membrane' models [5]. These techniques tend to be slow to converge and cumbersome to apply to practical problems. Also, this approach cannot directly help with problems such as transparency. In such situations every point has two motions, and no spatial segmentation can separate them.

Another approach avoids the issue of segmentation by relaxing the assumption that a single motion describes change within the region of analysis. This has been developed to deal with scenes containing several moving objects [4, 6, 11]. Approaches to this problem that are based on multiple peak detection in Hough Transform representations or cross-correlation functions [4, 6] generally lack robustness. Recently, the problem of multiple motion has been treated in the spatio-temporal domain for the case in which motion is uniform over many frames [11].

In this paper we introduce an alternative model for describing *local* motion in an image in which there may be two differently moving patterns within the neighborhood of an image point. This *two-component motion model* allows analysis of most basic local motion configurations (including transparent motion) that do not conform to the traditional single-motion model. Based on this model, we describe an algorithm for accurately estimating two motion components within a local analysis region using three frames of an image sequence.

2 Motion Configurations

Motion estimation at an image point is based on pattern information in a neighborhood of that point. We will refer to this neighborhood as the *motion analysis region.*

We have assembled in Figure 1 a small set of *elementary motion configurations* which can occur in a motion analysis region. These configurations are:

1. Single motion.
2. Two differently moving patterns separated by a boundary.
3. Two transparent surfaces with different motions. Examples include moving shadows, reflections, as well as actual transparent objects.

*Permanent address: Dept. of Computer Science, The Hebrew University of Jerusalem, 91904 Jerusalem, Israel.

CH2934-8/90/0000/0027$01.00 © 1990 IEEE

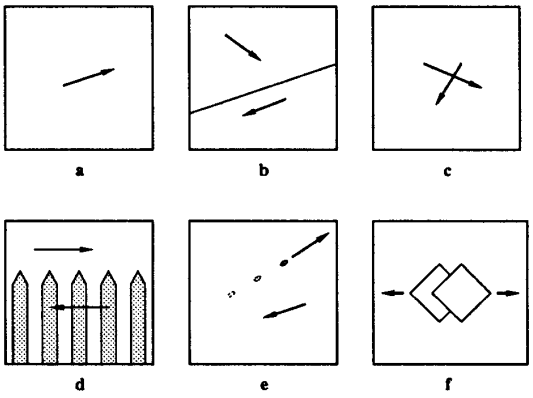

Figure 1: Elementary local motion configurations. For description, see text

4. Two interleaved components: small or thin foreground objects that move in front of a differently moving background, or the background appears through small gaps in the foreground.
5. A dominant moving pattern and a second pattern that has low contrast or is small. An example is a football partially tracked by the camera in a sports broadcast.
6. Aperture effect exists for each of two components.

3 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.

Let $I(x, y, t)$ be the observed grayscale 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 [7, 1] assumes that within the region R, $I(x, y, t)$ may be represented as a pattern $P(x, y)$ moving with velocity $\mathbf{p}(x, y)$.

$$\begin{aligned} I(x, y, 0) &= P(x, y), \\ I(x, y, 1) &= P(x - p_x, y - p_y) = P^{\mathbf{p}}, \\ I(x, y, t) &= P(x - tp_x, y - tp_y) = P^{t\mathbf{p}}, \end{aligned} \tag{1}$$

where $P^{t\mathbf{p}}$ denotes the pattern P transformed by the motion $t\mathbf{p}$ (see Figure 2a). This model can represent only the first of the elementary motion configurations in Figure 1 because it assumes that locally there is only one coherent motion.

We introduce an alternative model for local motion, as shown in Figure 2b. Within the analysis region the image is assumed to be a combination of two distinct image patterns, P and Q, having independent motions of $\mathbf{p}$ and $\mathbf{q}$:

$$I(x, y, 0) = P(x, y) \oplus Q(x, y), \tag{2}$$

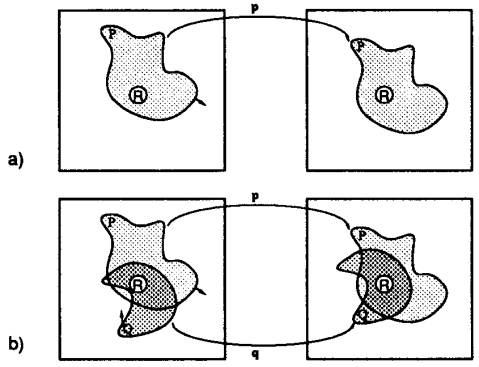

Figure 2: Two models for local motion.
a) The traditional model with single motion.
b) The two-motion model: two patterns, P and Q, move with velocities $\mathbf{p}$ and $\mathbf{q}$.

$$I(x, y, t) = P^{t\mathbf{p}} \oplus Q^{t\mathbf{q}}. \tag{3}$$

Here the $\oplus$ symbol represents an operator such as addition or multiplication to combine the two patterns.

With appropriate choices of patterns P and Q and of the combination operator $\oplus$, the proposed two-motions model can represent all of the elementary motion configurations shown in Figure 1. In transparent cases (Figure 1c), P and Q cover the analysis region and $\oplus$ is addition or multiplication. The more general foreground-background configurations shown in Figure 1 can sometimes be treated as *approximately* additive: they can be represented as additive over some subset of the analysis region. In what follows we will limit our consideration to the additive case, but we will show that an algorithm based on the assumption of additivity is robust with respect to violations of this assumption encountered in the more general cases.

4 Estimating a Single Motion

We now review an algorithm based on *coarse-fine tracking* for estimating a single image motion in accordance with the model of Equation (1). Various components of this algorithm for estimation of single motions have been described previously [1, 2, 3, 8], but their combined use has some important properties. In the next section we show that this procedure for estimating single-component motion can be applied repeatedly to extract two motion components.

For small displacements between frames $I(x, y, t-1)$ and $I(x, y, t)$ of an image sequence we can use the incremental motion estimator derived by Lucas and Kanade [8]. In general, this estimation method is accurate only when the frame-to-frame displacements due to motion are a fraction of a pixel, so that the truncated Taylor series approximation is meaningful. The precision of the estimate can be improved significantly through an iterative alignment procedure [8]. However, much better results can be obtained and the range of the motion estimation process can be ex-

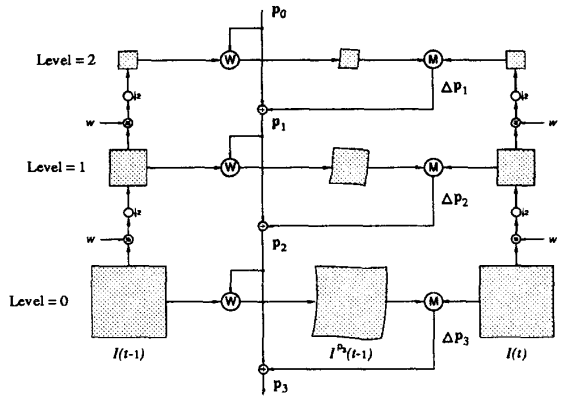

Figure 3: Diagram of the coarse-fine motion tracking algorithm.

tended to the general case of large displacements by implementing tracking within a multiresolution pyramid structure, Figure 3 [3].

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

Motion analysis begins at a low resolution level of the image pyramid where sample distance is large and correspondingly large image velocities can be estimated. At each successive iteration of the tracking procedure analysis moves to the next higher resolution pyramid level. Thus if level ℓ is processed at iteration k, and the motion estimate from the previous iteration is $\mathbf{p}_{k-1}$, a shift of $\mathbf{p}_{k-1}$ is applied to pyramid level $G_{t-1,\ell}$ to form $G_{t-1,\ell}^{\mathbf{p}_{k-1}}$. The residual motion, $\Delta\mathbf{p}_k$, is computed between this shifted image 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 the next higher resolution pyramid level. Thus coarse-fine tracking can efficiently estimate velocities of many pixels per frame time, at accuracies of a small fraction of a pixel [1, 2, 3].

5 Estimating Two Motions

We now consider the analysis of motion described by the two-component model. A 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 compute the other motion using the single-motion algorithm without making any assumptions about the nature of the patterns P and Q. In what follows we will assume that the combination operation is addition.

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

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

$$\begin{aligned} D_1 &\equiv I(x,y,2) - I^{\mathbf{p}}(x,y,1) \\ &= (P^{2\mathbf{p}} + Q^{2\mathbf{q}}) - (P^{2\mathbf{p}} + Q^{\mathbf{q}+\mathbf{p}}) \\ &= Q^{2\mathbf{q}} - Q^{\mathbf{q}+\mathbf{p}} = (Q^{\mathbf{q}} - Q^{\mathbf{p}})^{\mathbf{q}}, \end{aligned} \tag{3}$$

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

The sequence $\{D_n\}$ now consists of a new pattern $Q^{\mathbf{q}} - Q^{\mathbf{p}}$ moving with a single motion $\mathbf{q}$, that is: $D_n = (Q^{\mathbf{q}} - Q^{\mathbf{p}})^{n\mathbf{q}}$. Thus the motion $\mathbf{q}$ 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 $\mathbf{p}$ can be recovered when $\mathbf{q}$ is known. The observed images $I(x, y, t)$ are shifted by $\mathbf{q}$, and a new difference sequence is formed:

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

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

Note that the shift and subtract procedure removes, or "nulls," one moving pattern from the image sequence without determining what that pattern is, that is without explicit segmentation.

In practice, of course, neither motion $\mathbf{p}$ or $\mathbf{q}$ is known *a priori*. However, it is possible to recover both motions precisely if we start with even a very crude estimate of either. Two-component motion analysis can therefore be formulated as an alternating iterative refinement procedure. Let $\mathbf{p}_n$ and $\mathbf{q}_n$ be the estimates of motion after the n^{th} cycle. Refined estimates are obtained alternately for $\mathbf{p}$ and $\mathbf{q}$, so if $\mathbf{p}$ is obtained on even-numbered cycles, $\mathbf{q}$ is obtained on odd cycles. Steps of the procedure are:

1. Set initial estimate for the motion $\mathbf{p}_0$ of pattern P.
2. Form the difference images D_1 and D_2 as in Equation (3) using the latest estimate of $\mathbf{p}_n$.
3. Apply the single-motion estimator to D_1 and D_2 to obtain an estimate of $\mathbf{q}_{n+1}$.
4. Form new difference images D_1 and D_2 using the estimate $\mathbf{q}_{n+1}$.
5. Apply the single-motion estimator to the new sequence D_1 and D_2 to obtain an update $\mathbf{p}_{n+2}$.
6. 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 accurately after three to five cycles regardless of the initial guess of $\mathbf{p}_0$.

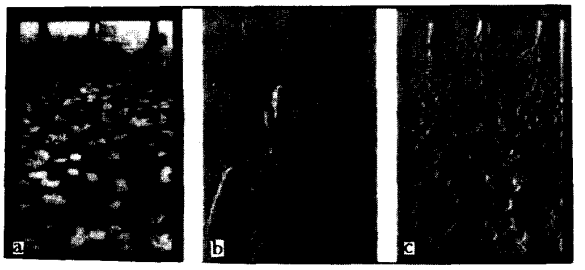

Figure 4: Registration of components in transparent motion, as described in text.

a) One frame from the sequence.
b) Difference of two consecutive frames after registration using first motion.
c) Difference of two consecutive frames after registration using second motion.

6 Examples

We have tested the two-motion algorithm with several examples of the elementary motion configurations shown in Figure 1. We have used both artificial sequences with known velocities, and real images of complex natural scenes. In all examples in this section, the analysis region R is taken to be the entire image and the images were of size 256×256 or 256×200 pixels.

Example 1: Transparent Motion

An example involving additive transparency is shown in Figure 4. 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 Figure 4a. As the camera moved, the image reflected in the glass and the image in the print moved differently. The algorithm was applied to recover the two motions. To demonstrate the accuracy of the computation, two consecutive frames are registered using each of the computed motions, and a difference image formed. A component will be canceled out in a difference image if registration is done with the motion of that component. Resulting difference images are shown in Figure 4b and Figure 4c. In Figure 4b the reflected image (barely visible in Figure 4a) is revealed showing that the other component was registered accurately. In Figure 4c, the reflected image has been nulled.

Example 2: Masking

A second sequence demonstrates motion recovery when one motion pattern predominates, and 'masks', the second pattern as in Figure 1e. 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 Figure 5a. 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

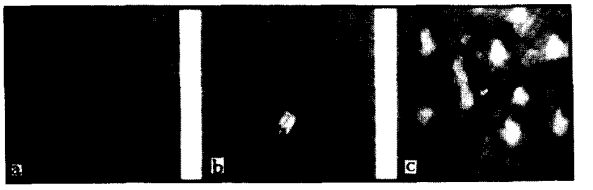

Figure 5: Registration of components with a small moving object against a large moving background.

a) One frame from the sequence. The toy tank and the background have different motions.
b) Difference of two consecutive frames after registration using the background motion.
c) Difference of two consecutive frames after registration using the tank motion.

foreground motions. The background cancellation is shown in Figure 5b and the foreground cancellation in Figure 5c. 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.

Example 3: Interleaved motion

The final example, Figure 6, shows an image sequence in which a crowd of people is viewed through a complex pattern of tree branches. This is an example of the configuration in Figure 1d. The camera is translating and rotating, so 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 rather than simple translations. In spite of many violations of the additivity assumption due to occlusion and exposure, convergence is reached after 4 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, Figure 6c and d. In each of these, the registered areas are sharp, while the rest of the image is blurred due to the image motion. For reference, an unregistered temporal average is shown in Figure 6b.

7 Stability Analyses

The examples shown in the preceding section suggest that the algorithm described is surprisingly robust with respect to violations of the assumptions expressed in Equation (2). Of the examples shown, only Example 1 involving 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 Example 6, 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.

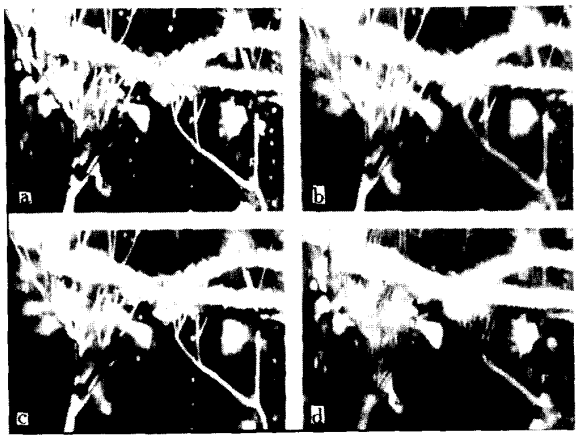

Figure 6: Registration of interleaved motion components.
a) One frame from the original sequence.
b) Averaging three consecutive frames from the original sequence. The entire scene is blurred.
c) Averaging three frames after registration with the foreground motion. The trees are sharp, while the background blurs out.
d) Averaging three frames after registration with the background motion. The background remains sharp, while the foreground blurs out.

7.1 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 per frame, one to the right, the other to the left.

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.

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 (10 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.

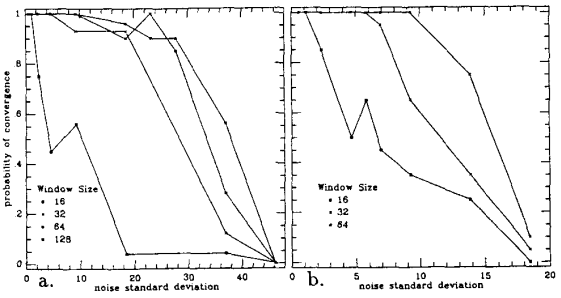

Figure 7: Probability of convergence as a function of noise level. For details see text.
a) Uncorrelated Noise: new samples of noise were generated for each frame.
b) Moving Noise: one sample of noise was generated, and then moved upwards by three pixels on each frame.

7.2 Results

Figure 7a shows the results using uncorrelated noise. On the abscissa is the standard deviation of the noise. Since the noise was uniformly distributed, the range of the noise is the standard deviation multiplied by 1.732. On the ordinate is shown the probability that the two-motions algorithm converged to within 20% of the correct velocities within 10 cycles of the 'estimate-subtract' analysis process. The error is defined as the rms error divided by the rms amplitude of the velocities, thus convergence requires that both motions be reasonably well estimated. 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 16×16, 32×32, 64×64, and 128×128 pixels.

With little or no noise, even a window size of only 16×16 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 3 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 32×32.

The results of the second experiment are shown in Figure 7b. A third motion component is introduced, moving at the same speed as the original two, (3 pixels per frame), but moving upward rather than right or left. The axes are as in Figure 7a. For the 16×16 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

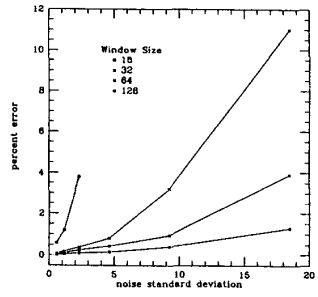

Figure 8: Percentage RMS error when probability of convergence is above 50%.

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 Figure 8, which shows the RMS deviation of the estimated velocities from the true values for the cases in which convergence was obtained. 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 and never gets above 10%. Similar precision is found in the case of the moving noise when conditions yielding similar probabilities of convergence and window sizes are compared.

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-motions model will rarely be satisfied precisely.

8 Concluding Remarks

A method has been presented for detecting two components of motion within an image region using three frames. This technique is based on a two-component model of local image motion, which is a generalization of the single-component model implicit in standard optical flow computation. The technique does not require segmentation to obtain precise motion estimates. Instead, it relies on an iterative multiresolution tracking process in which each estimate of one component of the motion is used to improve the accuracy of the other. This allows the motions to be estimated accurately without explicitly knowing their corresponding pattern components.

Advantage can be taken of the proposed method to improve optical flow computations. Current approaches to flow estimation are forced to use small neighborhoods in the computation of each motion vector so that the likelihood that a neighborhood will overlap a motion boundary is small. Multiple motion analysis as proposed here, allows larger neighborhoods to be used, since neighborhoods can overlap motion boundaries without violating assumptions of the analysis. Larger regions lead, in turn, to more precise and robust motion estimates.

References

[1] P. Anandan. A unified perspective on computational techniques for the measurement of visual motion. In *International Conference on Computer Vision*, pages 219–230, London, May 1987.

[2] J.R. Bergen and E.H. Adelson. Hierarchical, computationally efficient motion estimation algorithm. *J. Opt. Soc. Am. A.*, 4:35, 1987.

[3] P.J. Burt, J.R. Bergen, R. Hingorani, R. Kolczinski, W.A. Lee, A. Leung, J. Lubin, and H. Shvaytser. Object tracking with a moving camera, an application of dynamic motion analysis. In *IEEE Workshop on Visual Motion*, pages 2–12, Irvine, CA, March 1989.

[4] C.L. Fennema and W.B. Thompson. Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing*, 9:301–315, 1979.

[5] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:721–741, November 1984.

[6] B. Girod and D. Kuo. Direct estimation of displacement histograms. In *Image Understanding and Machine Vision*, pages 73–76, Cape Cod, June 1989. Optical Society Of America.

[7] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[8] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130, 1981.

[9] D.W. Murray and B.F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(2):220–228, March 1987.

[10] S. Peleg and H. Rom. Motion based segmentation. In *International Conference on Pattern Recognition*, volume 1, pages 109–113, Atlantic City, June 1990.

[11] M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *International Conference on Pattern Recognition*, volume 1, pages 274–278, Atlantic City, June 1990.