Accurate Motion Flow Estimation with Discontinuities*

Lionel Gaucher and Gérard Medioni
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089-0273
{lgaucher,medioni}@iris.usc.edu

Abstract

We address the problem of motion flow estimation for a scene with multiple moving objects, observed from a possibly moving camera. We take as input a (possibly sparse) noisy velocity field, as obtained from local matching, produce a set of motion boundaries, and identify pixels with different velocities in overlapping layers. For a fixed observer, these overlapping layers capture occlusion information. For a moving observer, further processing is required to segment independent objects and infer structure. Unlike previous approaches, which generate layers by iteratively fitting data to a set of predefined parameters, we instead find boundaries first, then infer regions and address occlusion overlap relationships. All computational steps use a common framework of ***tensors*** *to represent velocity information, together with saliency (confidence), and uncertainty. Communication between sites is performed by convolution-like tensor* ***voting****. The scheme is non-iterative, and the only free parameter is the scale, related to neighborhood size. We illustrate the approach with results obtained from synthetic sequences and from real images. The quantitative results compare favorably with those of other methods, especially in the presence of occlusion.*

1 Introduction

We seek to determine accurate optical flow from a motion sequence. Early methods have relied on local, raw estimates of the optical flow field to produce a partition of the image. This leads to severe limitations, as the flow estimates are known to be very poor at boundaries, and cannot be obtained in uniform areas. In addition, the calculation of optical flow is a coupled problem. The determination of accurate flow requires prior knowledge of discontinuities at motion boundaries where smoothness constraints must be relaxed. But locating discontinuities presupposes knowledge of the flow.

Past methods have investigated the usefulness of Markov Random Fields (MRF) in treating discontinuities in the optical flow[12]. Regularization techniques which preserve discontinuities by weakening the smoothing of areas which demonstrate strong intensity gradients have also been used[13]. More recently, significant improvements have been achieved by casting the problem in terms of layered descriptions[1][2][3][4]. This novel formalism has many advantages. It is a natural way to accomodate discontinuities present in the motion field. Also, it allows information transfer between spatially separated regions, and may resolve local uncertainties.

But the actual mapping of pixels to layers is difficult. Many current methods use common motion to group regions, usually performing a parameterized fit to motion data[5][6]. Weiss[7] provides a good overview of the difficulties involved in this estimation process, which range from inadequate representation of motion as rigid and non-planar, to the overfitting and instabilities resulting from higher-order parameterizations.

Weiss performs image segmentation using a variant of the Expectation-Maximization (EM) algorithm[8], where a dense smooth flow field is fit to multiple layers. But methods dependent strictly upon a mathematical fitting can be limited by a lack of higher-level analysis. It is possible for unrelated regions to be accidentally merged into a single layer simply because of similar motion profiles, despite the presence of conflicting evidence (e.g. occlusion). The merging of spatially diffuse regions is more appropriately the domain of higher-level processing.

Within the same layered description framework, we present here a completely different approach in which we first detect boundary elements between smooth velocity fields. We then locally group these into curves. The velocity fields near these boundaries are then refined.

The determination of motion boundaries prior to the refinement and smoothing of the velocity field bordering these boundaries effectively decouples the problem of determining accurate optical flow. After refining the boundaries and the velocity fields near them, occlusion relationships between regions are determined. Pixels with different velocities in separate layers are easily identified.

All of the computational steps, from local boundary detection to velocity refinement, are implemented in a common framework which involves voting and tensor calculus[15]. This non-linear methodology is non-iterative, does not depend upon critical thresholds, and is robust in the presence of local irregularities in the motion field.

Section 2 presents an overview of the methodology, as well as a flowchart illustrating the algorithm. Section 3 presents the proper background for understanding the ten-

*This research is supported by contract DAAB07-97-C-J023, funded by DARPA, and monitored by U.S. Army, Fort Monmouth, NJ.

sor voting formalism which is used throughout the study. Section 4 describes the acquisition of the initial velocity input. The next four sections present the details of the steps. Section 9 shows results of the method on motion sequences, and Section 10 presents conclusions.

2 Our Approach

Figure 1 illustrates the steps of our method. The input is a field of velocity vectors, derived via a three-frame maximum cross-correlation technique. We then generate a dense tensor velocity field, which encodes not only velocity, but also estimates of confidence (saliency) and uncertainty. We then extract discontinuities from this field, which are found as locations of maximal velocity uncertainty using the tensor voting formalism. Interpreting these uncertain velocity locations as local estimates of boundaries of regions, tensor voting is used again to both align tangents along these boundaries, and to join these tangents into region boundaries.

Having segmented the motion field, tensor voting is used again between pixels not separated by boundaries to accurately estimate the velocities at the borders of these objects (which are inherently uncertain in the presence of occlusion).

With coherent velocities at the borders of these objects, a *local* representation of occlusion is found by determining which region's velocity field dominates in both future and past frames. From this analysis, the locations of pixels with multiple velocities are determined.

3 Tensor Voting and Saliency

We propose to augment the traditional representation of local information (here, a displacement vector) by two critical components, *saliency* which expresses the degree of confidence associated with the measurement, and *uncertainty*. This compound information can be conveniently expressed by an ellipsoid (ellipse in 2-D), where the *shape* of the ellipsoid conveys the direction and uncertainty, and its absolute *size* expresses saliency. Mathematically, it is known as a (second-order, symmetric) tensor[10].

A useful statistical representation of an ellipsoid is as the covariance matrix $\boldsymbol{S}$ derived from a distribution of points on its surface.

$$\boldsymbol{S} = \begin{bmatrix} \hat{\boldsymbol{e}}_1 & \hat{\boldsymbol{e}}_2 & \hat{\boldsymbol{e}}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{e}}_1^T \\ \hat{\boldsymbol{e}}_2^T \\ \hat{\boldsymbol{e}}_3^T \end{bmatrix} \qquad (1)$$

The eigenvalues $\lambda_1, \lambda_2, \lambda_3$ (where $\lambda_1 \geq \lambda_2 \geq \lambda_3$) correspond to each of the principal directions $\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3$. The eigenvalues determine the shape of the ellipsoid while the eigenvectors determine the orientation (Figure 2). Since

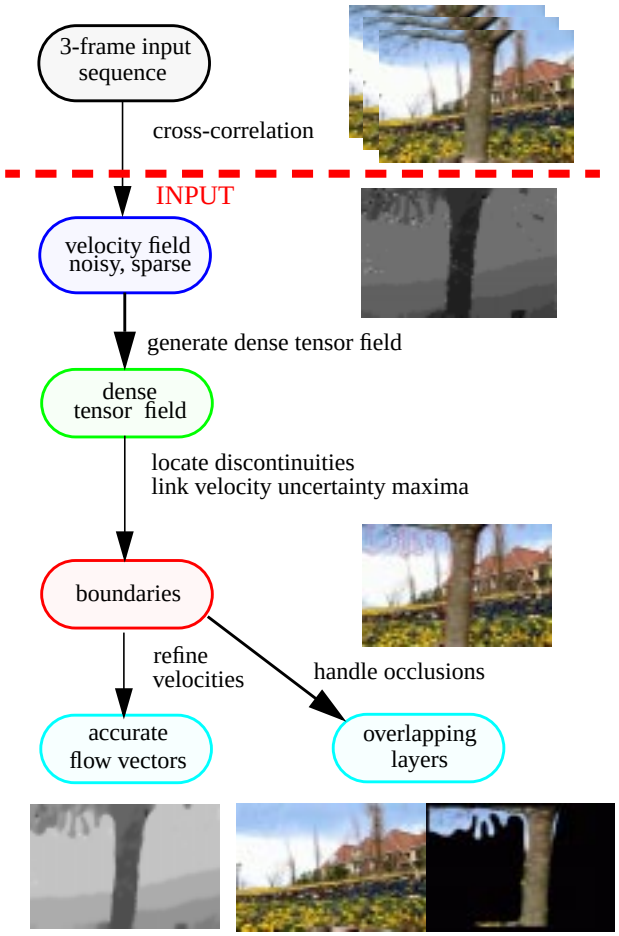


Figure 1 Determination of Image Layers

the eigenvalues determine the shape and size of the ellipsoid, they also convey saliency and uncertainty information.

A simple rearrangement of (1) yields the following:

$$\boldsymbol{S} = (\lambda_1 - \lambda_2)\hat{\boldsymbol{e}}_1\hat{\boldsymbol{e}}_1^T + (\lambda_2 - \lambda_3)(\hat{\boldsymbol{e}}_1\hat{\boldsymbol{e}}_1^T + \hat{\boldsymbol{e}}_2\hat{\boldsymbol{e}}_2^T) + \lambda_3(\hat{\boldsymbol{e}}_1\hat{\boldsymbol{e}}_1^T + \hat{\boldsymbol{e}}_2\hat{\boldsymbol{e}}_2^T + \hat{\boldsymbol{e}}_3\hat{\boldsymbol{e}}_3^T) \qquad (2)$$

The first term represents a "stick" component of the saliency tensor $\boldsymbol{S}$, with complete dominance by a single orientation. The second and third terms represent "plate" and "ball" components. In the plate component, two equal eigenvalues co-dominate. In the ball component, all eigenvalues are equal; no orientation is favored.

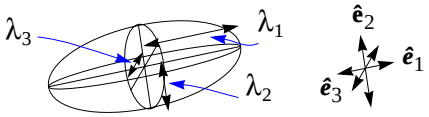


Figure 2 Ellipsoid and Eigensystem

Given a (possibly sparse and noisy) set of velocity vectors as input, we can generate a dense tensor field by al-

lowing active sites to communicate with their neighbors. This communication is performed by a convolution-like operation, and produces a tensor at every location.

It is necessary to provide a voting function ***V(S,p)*** which provides the value of the tensor field for a saliency tensor ***S*** at a location ***p*** relative to the tensor's coordinate system. The strength of the field should decrease with distance and be orientation-independent.

In addition, the linear nature of the voting field allows us to exploit the component expansion of ***S*** given above to provide fields for the stick, plate, and ball components. Some linear combination of these is sufficient to represent any saliency tensor. Furthermore, the orientation-independence of the field allows each of the three fields to be calculated once and stored for all future uses. Application can then be in the form of a convolution mask properly oriented to suit the originating saliency tensor's principal axis.

The functional form of the ball field used in this work is $V(S, p) = \exp(-p^2/\sigma^2)\hat{p}\hat{p}^T$, where σ is a scale factor. This functional form obviously satisfies the symmetry and decay requirements of the ball field. The stick field used is the same 2-D extension field of Guy and Medioni[9][14], whose work provides detailed functional forms.

Following tensor voting, the eigenvalues and eigenvectors of the saliency tensor at each voted site are determined. The saliency tensor at the recipient site can then be divided into the stick, plate, and ball components. Accordingly, the saliency of each of these is (λ_1-λ_2), (λ_2-λ_3), and λ_3 respectively. Features corresponding to each of these components are then located at the local maxima of the corresponding saliency, and extraction is performed by non-maximal suppression on the feature saliency map[14]. (For example, local extrema of the plate tensor correspond to surface discontinuities.)

4 Velocity Field from Three Frames

The raw velocity field which is provided as input should be as accurate as possible. In recent work[15], a standard two-frame maximum cross-correlation coefficient technique was used. While this two-frame technique gives adequate values for the motion field where velocities vary slowly, areas in which differently moving objects are simultaneously found within the convolution mask are more troublesome. Even worse are the results in areas of the first frame which are about to be occluded in the second frame, since there can be no meaningful correlation detected in this case. In these areas in particular, it is very difficult to determine either the correct motion field or the proper boundaries between objects.

Making a few reasonable assumptions about the nature of the observed object motion suggests that a cross-correlation calculation in which *three* consecutive frames are used, leads to more accurate results. It can generally be assumed at the *local* level that most occlusion events involve only two conflicting motion boundaries. Further assuming that the objects in a scene are locally convex and demonstrate negligible acceleration between frames, one can conclude that an area at time t which is about to be occluded at time $t + \Delta t$, was probably also visible at time $t - \Delta t$. In other words, an object which is being occluded in forward time is likely to be in the process of being uncovered (disoccluded) in reverse time.

Since disoccluding events pose less trouble than occluding events during determination of the motion field, this suggests that a more accurate estimate of the local velocity can be attained by choosing the best cross-correlation match in either forward or reverse time, negating the velocity in the case where the reverse time cross-correlation is larger.

Since the correlation mask has finite extent, there will still be weaker cross-correlation where an object boundary crosses the mask. But, most importantly, these areas of weak cross-correlation are now roughly *symmetrically* distributed around the true motion boundary, rather than being considerably more extended into the object undergoing occlusion in forward time. This enables the tensor-voting formalism to more accurately locate the motion boundary. The cross-correlation coefficient also offers a measure of strength to be used in the tensor-voting process.

5 From Velocity Field to Tensor Field

The first step of the process is to convert the input flow field into a dense tensor field. Figure 3(a) shows a frame from the "Flower Garden" sequence, and Figure 3(b) shows the horizontal and vertical components of the input velocity field. Note that velocities near the motion boundaries are incoherent and the boundaries are irregular.

At each point, the displacement vector $(v_x\ v_y)^T$ between P_t and P_{t+1} is the projection onto the xy plane of the 3-D vector $(v_x\ v_y\ \Delta t)^T$. Assuming the sampling rate is constant (and set to 1), this flow can be represented by two variables. As explained in Section 3, we want to encode saliency as the size of the tensor, so we map the velocity vector $(v_x\ v_y)^T$ to $(v_x\ v_y\ 1)^T$, but scaled down to a *unit* vector. Note that such a representation does not introduce any motion bias, and that the null flow maps to the unit vector $(0\ 0\ 1)^T$. Similarly, given a tensor with a long axis given by $(a\ b\ c)^T$, the length $\sqrt{a^2 + b^2 + c^2}$ represents the saliency, and the corresponding image velocity is $(a/c\ b/c)^T$.

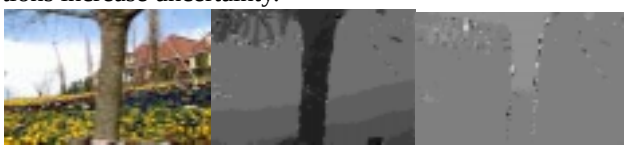
This technique, which represents velocity vectors of varying lengths as unit vectors in a higher dimensional space, prevents high velocities from disproportionately in-

fluencing the tensor-voting process. The weight of the unit vectors can be modulated by a confidence measure.

5.1 Initial Vote

We now allow all the sites with velocity information to communicate with each other, and with empty sites. This is performed as a convolution with a ball field, which is the simple scaled Gaussian field already described.

Intuitively, each site broadcasts its current motion to its neighbors, but allows deviations from it.The result is a true tensor field, encoding velocity information, saliency, and uncertainty. Adjacent sites with similar motion increase saliency, whereas adjacent sites with different motions increase uncertainty.



(a) A Frame (b) Input X- and Y-velocities

Figure 3 Flower Garden Sequence

6 Segmentation of the Motion Field

6.1 General Description

Assuming that the interiors of moving regions exhibit smoothly varying velocity field values, boundaries between moving objects can be detected by extracting curves of relative maxima in the *uncertainty* of the velocity. These areas of maximal uncertainty result from the fact that boundaries between neighboring regions with different velocities are influenced by both of these regions during voting.

6.2 Regions of Maximally Uncertain Velocity

Following our first vote, and diagonalization of its covariance matrix, each tensor is then characterized by a principal axis (representing an encoded velocity), and eigenvalues λ_1, λ_2, and λ_3, where $\lambda_1 \geq \lambda_2 \geq \lambda_3$. We use as a measure of velocity uncertainty the quantity λ_2/λ_1, which will approach unity as uncertainty increases. (See Figure 4a.)

This uncertainty measure varies smoothly across the image. Relative maxima in the uncertainty will occur along "ridges" which represent boundaries between regions of differing velocity. These locations are found by a modified version of the Marching Square algorithm[9][11].

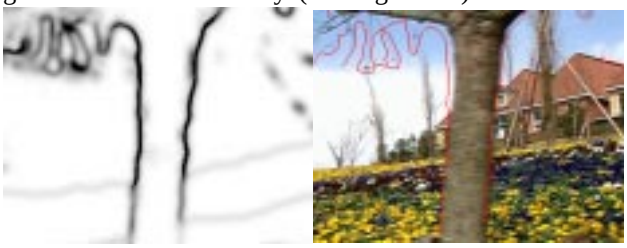
6.3 Determination of Component Boundaries

These boundary curves of maximally uncertain velocity lie between regions of differing velocities. These curves are later used to determine which pairs of pixels may communicate during a velocity refinement procedure. It is therefore advantageous to complete these boundaries to the greatest extent possible by finding the most likely curves passing through these regions.

First, we assign a tangent to the pixels in these maximally uncertain regions. The 2-D extension field[9][14] is ideally suited for this purpose. At each pixel judged to be maximally uncertain, other such pixels vote for prospective tangents. These tangents are derived from unit vectors parallel to segments joining the voting pixel to the recipient pixel. Voting is restricted to maximally uncertain pixels, resulting in a sparse 2-D tensor field. The principal axis of the pixel's diagonalized covariance matrix determines the resultant tangent direction. The strength of the tangent vote is taken to be the magnitude of the stick component of the 2-D tensor, $\lambda_1 - \lambda_2$.

The result of this 2-D convolution-like operation is a dense 2-D field of 2-D tensors (ellipses) where the shape represents uncertainty and the size saliency. We extract curves from the dense field as maxima of the stick component, once again using a modified Marching Square procedure.

These edges represent the boundaries of the desired regions. The velocity field is therefore segmented into regions of coherent velocity. (See Figure 4b.)



(a) Uncertainty Map (b) Region Boundaries

Figure 4 Determination of Boundaries

7 Region Refinement

7.1 General Description

With the initial segmented description now complete, we go from a pixel-level representation to a region-level representation. A *local* determination of occlusion between regions will be made based upon velocities present near region boundaries. In the presence of an occlusion event, however, these velocities are the most uncertain. A more elaborate local analysis is therefore necessary.

7.2 Region-Level Velocity Refinement

Near the boundary between two regions moving differently, the velocity information is necessarily inaccurate and corrupted, as it is estimated from a mixture of velocities. Furthermore, occlusion of a region by another in time also alters the true velocity in the occluded area. Now that we have boundaries between regions, we can overcome

these problems by another round of tensor voting, with some slight changes.

In this round, voting is only permitted between pixels which can be connected with a straight line which does not cross a region boundary. And the strength of a pixel's velocity vote is proportional to $1 - \lambda_2/\lambda_1$, where λ_1 and λ_2 are the eigenvalues resulting from the diagonalization of that pixel's covariance matrix during first stage voting.

The quantity $1 - \lambda_2/\lambda_1$ is a measure of the certainty of that pixel's velocity. The more certain velocities of the region supplant the less certain ones near the region boundaries. This has the effect of refining the velocity field within each region, and compensating for a lack of reliable velocity information near region boundaries.

It should be noted that the refined velocities near region boundaries are still *locally* influenced, and are not averaged over the entire region. This allows for accurate representation of objects which exhibit variations in velocity, such as rotating or slowly deforming objects.

Results of the velocity refinement as applied to the Flower Garden sequence are shown in Figures 5(a) and 5(b), the horizontal and vertical components, respectively, of the refined velocity field.

(a) Refined X-velocities (b) Refined Y-velocities

Figure 5 Refined Velocity Field

At this level of processing, we have provided a higher level of description for the image which preserves discontinuities in the motion field at region boundaries, but still permits further refinement within individual regions.

In this way, we have effectively circumnavigated the coupled nature of the problem of solving for the optical flow. Restricting the tensor voting to occur on only one side of a region boundary allows refinement of the velocity field subject to the boundary conditions imposed by the presence of discontinuities at the region borders.

Despite the presence of smoothly coherent velocity fields within these regions, no attempt is made at this point to partition the set of regions into meaningful objects. This process requires determination of other higher-level relationships between regions. By postponing the merging of regions until further information (e.g. occlusion, or even higher-level semantic relationships) is computed, the methodology avoids the pitfalls of relying on a low-level mathematical fit for determining when regions can be merged.

8 Handling Occlusion

8.1 General Description

At this point, the motion field has been refined and the uncertain velocities near the component boundaries have been replaced by more accurate estimates. Using the refined velocity field at time t_0, and assuming the absense of any occluding components, the velocity field at time $t_1 = t_0 + \Delta t$ can be predicted. Region pixels simply translate in time to their new positions.

But in the presence of occlusion, the velocity field at time t_1 will depend upon which regions at time t_0 occlude others. When pixels from two regions at time t_0 are predicted to project into the same location at time t_1, the occluding region will determine the velocity at the new location. Therefore, to the extent that two "conflicting" pixels in separate t_0 regions differ in their velocities, the refined velocity field at time t_1 can be used to determine occlusion orderings between the regions at time t_0. It should be noted that the availability and reliability of such time-projected conflict information depends heavily upon the accuracy of the velocity refinement process previously described.

Unfortunately, such predictive capability does not exist with boundaries which *uncover* regions. The portion of a region occluded at time t_0 cannot predict a velocity at time t_1 since velocities in the t_0 occluded region are not available. Since resolution of this velocity conflict at time t_1 is necessary to determine the nature of the occlusion, no occlusion ordering information can be gained in this case. (See Figure 6.)

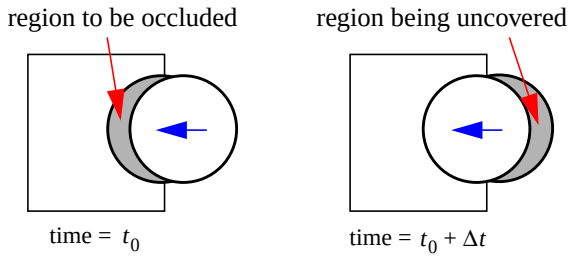


Figure 6 Uncertain Occlusion Boundaries

But occlusion classifications are invariant to time-reversal, and an uncovering event in forward time becomes an occlusion event in reverse time. Therefore, detection of occlusion in both forward and reverse time flow detects all occlusion events.

8.2 Criteria for Classification of Occlusion

We detect occlusion *locally* as follows. A *first* pixel is propagated from the previous (future) frame forward (backward) using the refined velocity field in that frame. This new *second* pixel in the present frame is then propa-

gated back, using its refined velocity, to the previous (future) frame to arrive at a *third* pixel.

If the *second* pixel in the present frame has *not* just occluded a pixel from another layer in either forward or reverse time, the *first* pixel will be the same as the *third* pixel in both cases (or at least they will not be separated by a motion boundary). Otherwise, in either forward or reverse time, the *first* pixel will be separated from the *third* pixel by a motion boundary. This allows us to locate pixels in the present frame which have dual values. The two velocities are easily determined, as is the order of the occlusion based on the refined velocity of the *second* pixel.

Figure 7 shows which pixels in the central frame of the Flower Garden sequence are dual valued. Clearly, this procedure depends heavily on having accurate (or at least consistent) placement of motion boundaries. Using the partial

Figure 7 Dual-velocity Pixels

order of occlusion derivable from this data, a separation into layers can be effected. Propagating background pixel velocities from several frames allows reconstruction of the background image, shown in Figure 8. Figure 9 shows a

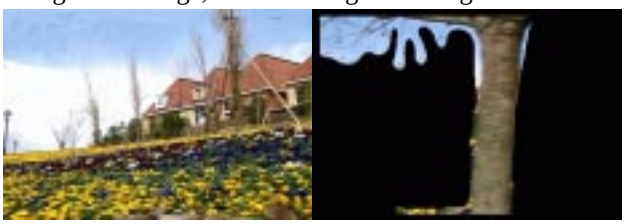


Figure 8 Flower Garden Layers

segmentation of a random dot motion sequence into overlapping layers. The accuracy of the segmentation is resistant to gradual distortions of the component objects.

Figure 9 Random Dot Motion Sequence

9 Additional Results

In addition to the previously shown results from the analysis of the Flower Garden sequence, we also present results from the analysis of five other three-frame sequences.

First, in order to demonstrate an ability to accurately obtain optical flow in regions undergoing some distortion, Figure 10 shows the results obtained from analysis of a synthetic sequence in which a disk composed of random dots undergoes expansion in front of a similarly-textured background. The disk border moves radially at 5 pixels/frame.

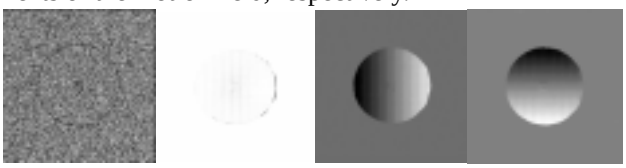
Figure 10(a) shows the second-frame disk in a three-frame sequence after final segmentation. Figure 10(b) shows the error in the refined velocity field, where darkness grows linearly with error.

Here, the error measure used is the "angular" error measure used by Barron, Fleet, and Beauchemin[16]. A velocity $\hat{v} = (v_1, v_2)^T$ is represented as the 3-D unit vector

$$\hat{v} \equiv (v_1, v_2, 1)^T / (\sqrt{v_1^2 + v_2^2 + 1})$$

in space-time coordinates. A 2-D velocity is then completely characterized by the orientation of this unit vector. The error measure used is $\theta_{error} = cos^{-1}(\hat{v}_c \bullet \hat{v}_e)$ where $\hat{v}_c$ is the correct velocity and $\hat{v}_e$ is the estimated velocity.

The average error found for the expanding disk is $4.05° \pm 5.85°$ for full 100% field coverage. Figures 10(c) and 10(d) show the refined horizontal and vertical components of the motion field, respectively.



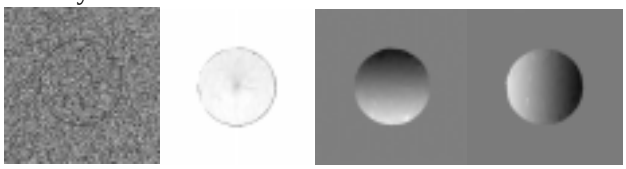
(a) (b) (c) (d)

Figure 10 Focus of Expansion Analysis

The velocities near the boundaries of the disk have been faithfully reproduced by the refinement voting. Distortion resulting from dissimilar rates of expansion have little effect on the refined velocity field. The method has no bias toward *constant* velocity motion in the image plane.

Figure 11 shows a similar analysis for a disk rotating counter-clockwise at approximately $12°$ per frame. In this case, the measured "angular" error is $8.80° \pm 13.8°$ for full 100% field coverage. The area near the center of rotation provides a weak correlation since its motion cannot be approximated linearly. Some error is also incurred by virtue

of the fact that rotation necessarily includes acceleration between frames. But, in particular, boundaries are very accurately found.

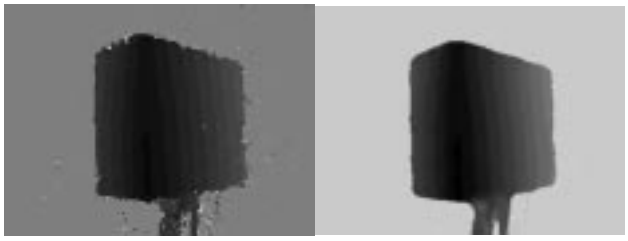


(a) (b) (c) (d)

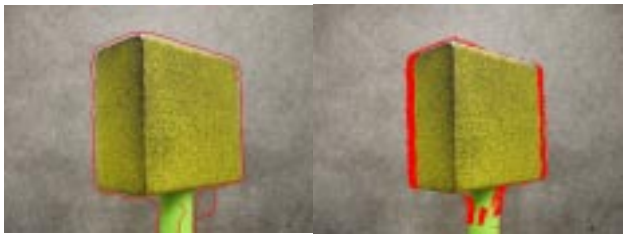
Figure 11 Rotational Analysis

Figure 12 shows three frames from a sequence in which a block mounted on a post is allowed to translate and rotate in front of a speckled background. The analysis is performed on the central frame. Figure 13(a) shows the horizontal component of the initial noisy velocity field. Figure 13(b) shows the scaled horizontal component of the motion field after the refinement voting procedure. The local nature of the tensor voting procedure easily accomodates variations in velocity along the border of the block resulting from its rotation.The accuracy of the edge placements and refined velocity field makes possible a realistic representation of occlusion in the scene..

Figure 12 Rotating Block Sequence



(a) Input X-velocities (b) Refined X-velocities

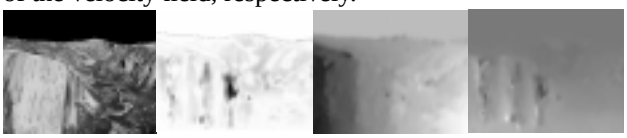


(c) Boundaries (d) Dual-velocity Pixels

Figure 13 Rotating Block Analysis

Figure 13(c) shows the resulting boundaries derived from the uncertainty map, superimposed on the original image. The boundaries accurately reflect the true motion boundary of the block, except at the top of the block where a lack of texture in a portion of the background has caused this portion to be merged with the block. Figure 13(d) shows the occlusion analysis applied to the rotating block sequence. The dual-velocity pixels are accurately placed due to the precision of edge determination and velocity refinement.

An analysis of three frames of the Yosemite sequence (without sky) is shown in Figure 14. Figure 14(a) shows the central frame of the three-frame subsequence used. Figure 14(b) shows the "angular" error map. The average error obtained is $8.83° \pm 10.6°$ for 100% field coverage, and $2.12° \pm 0.92°$ for 34% field coverage. Figures 14(c) and 14(d) show the refined horizontal and vertical components of the velocity field, respectively.



(a) (b) (c) (d)

Figure 14 Yosemite Analysis

The Yosemite sequence calculation was performed with only three frames of the sequence, but it could have been performed with only *two* since there is no appreciable occlusion present. Since the strength of this methodology is its ability to treat sequences presenting a substantial degree of occlusion, performance on this sequence does not completely convey the power of the technique.

Table 1

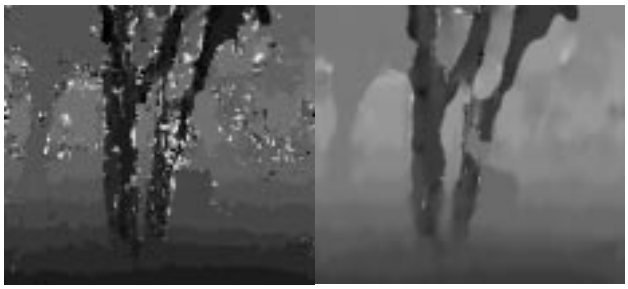
Sequence	Error (degrees)	Density
Expanding Disk	4.05 +/- 5.85	100%
Expanding Disk	2.32 +/- 0.87	70%
Expanding Disk	1.54 +/- 0.48	32%
Rotating Disk	8.80 +/- 13.8	100%
Rotating Disk	4.45 +/- 2.18	66%
Rotating Disk	2.81 +/- 1.35	37%
Yosemite	8.83 +/- 10.6	100%
Yosemite	3.71 +/- 2.07	61%
Yosemite	2.12 +/- 0.92	34%

Table 1 presents an error analysis for the sequences studied which have available ground truth data. It reports "angular" error for specific levels of coverage of the motion field. These results compare very favorably with those in the current literature[16].

As another example, the SRI Tree sequence is analyzed. Figure 15 shows the three frames used in the analysis. Figure 16(a) shows the horizontal component of the noisy input velocity field, while Figure 16(b) presents the same component after refinement. . With the exception of the admittedly more difficult lower half of the foreground tree, the boundaries and velocities derived in the upper half

are fairly accurate. Incorporation of monocular data in the analysis would obviously improve the results.

Figure 15 SRI Tree Sequence



(a) Input X-velocities (b) Refined X-velocities

Figure 16 SRI Tree Analysis

10 Conclusions and Future Work

We have presented some preliminary results of a novel methodology to address the issues of accurate optical flow computation using motion information *only.* It explicitly addresses the classical limitation that velocity information is necessarily inaccurate around motion boundaries, and that pixels may have multiple velocities.

Most importantly, it effectively demonstrates an ability to simultaneously determine motion boundaries and accurate optical flow without resorting to iterative global optimization techniques. This ability can be viewed as an important foundation upon which higher levels of image sequence processing can be based.

While these preliminary results are very encouraging, there is considerable room for improvement. For example, the stability of the method can be greatly improved by incorporating the coherence which exists between frames. All results presented here are obtained with only *three* frames.

In addition, the localization of motion boundaries can be made more accurate by the inclusion of monocular information (e.g. edges). This is particularly true for motion boundaries between occluding/occluded pairs in which the only difference between velocities on both sides of the boundary is an out-of-plane projection (e.g. boundaries of non-translating rotating objects).

Also, additional investigation is needed to determine how to combine information acquired at the local level (motion boundaries, occlusion evidence, and, eventually, edges) into a complete partitioning of the image into individual regions with coherent velocity. This will likely require merging of information sources with very different characteristics.

Further study must also be undertaken to determine criteria for grouping partitioned regions with similar motion profiles into the same layer. This process, which is usually performed in other techniques as the result of a mathematical fit at the pixel level, is more properly performed at a higher level of processing where characteristics of macroscopic entities (e.g regions, etc.) can influence the outcome. These are the topics of our ongoing research.

11 References

[1] T. Darrell and A. Pentland, “Robust estimation of a multilayered motion representation”, *Proc. IEEE Workshop on Visual Motion*, Princeton, 1991, pp. 173-178.

[2] A. Jepson and M. J. Black, “Mixture models for optical flow computation”, *Proc. IEEE Conf. Comput. Vision Pettern Recog.*, New York, 1993, pp. 760-761.

[3] S. Hsu, P. Anandan, and S. Peleg, “Accurate computation of optical flow by using layered motion representation”, *Proc. 12th Int’l Conf. Pattern Recog*, 1994.

[4] S. Ayer and H.S. Sawhney, “Layered representation of motion video using robust maximum likelihood estimation of mixture models and MDL encoding”, *Proc. Int’l Conf. Comput. Vision*, 1995, pp. 777-784.

[5] M. Irani and S. Peleg, “Image sequence enhancement using multiple motions analysis”, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Champaign, Illinois, 1992, pp. 216-221.

[6] J. Y. A. Wang and E.H. Adelson, “Representing moving images with layers”, *IEEE Trans. on Image Processing Special Issue: Image Sequence Compression*, Sept. 1994, 3(5): pp. 625-638.

[7] Y. Weiss, “Smoothness in Layers: Motion segmentation using nonparametric mixture estimation”, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Puerto Rico, 1997, pp. 520-526.

[8] A. P. Dempster, N.M. Laird, and D.B. Rubin “Maximum likelihood from incomplete data via the EM algorithm”, *J.R. Statist. Soc. B*, 39:1-38, 1977.

[9] G. Guy and G. Medioni, “Inferring Global Perceptual Contours from Local Features”, *IJCV*, vol. 20, no. 1/2, Oct 1996, pp. 113-133.

[10] G. H. Granlund and Knutsson, *Signal Processing for Computer Vision*, Kluwer Academic Publishers, 1995.

[11] W. E. Lorensen and H.E. Cline, ‘’Marching Cubes: A High Resolution 3-D Surface Reconstruction Algorithm’’, *Computer Graphics*, vol. 21, no. 4, July, 1987.

[12] F. Heitz and P. Bouthemy, “Multimodal Estimation of Discontinuous Optical Flow Using Markov Random Fields”, *PAMI*, vol. 15, no. 12, Dec. 1993, pp. 1217-1232.

[13] S. Ghosal, “A Fast Scalable Algorithm for Discontinuous Optical Flow Estimation”, *PAMI*, vol. 18, no. 2, Feb. 1996, pp. 181-194.

[14] G. Guy and G. Medioni, “Inference of Surfaces, 3-D Curves and Junctions from Sparse, Noisy 3-D Data”, *PAMI*, vol. 19, no. 11, Nov. 1997, pp. 1265-1277.

[15] L. Gaucher, G. Medioni, and J. Wilson, “Accurate Motion Flow Estimation Using a Multilayer Represenation”, *CVPR Workshop on the Interpretation of Visual Motion*, June 22, 1998, Santa Barbara.

[16] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of Optical Flow Techniques”, *IJCV*, vol. 12, no. 1, February 1994, pp. 43-77.