Exploiting Discontinuities in Optical Flow

William B. Thompson

Department of Computer Science
University of Utah
Salt Lake City, UT 84112

University of Utah Dept. of Computer Science Tech. Report UUCS-95-015, October 19, 1995, revised April 20, 1998.

ABSTRACT

Most optical flow estimation techniques have substantial difficulties dealing with flow discontinuities. Methods which simultaneously detect flow boundaries and use the detected boundaries to aid in flow estimation can produce significantly improved results. Current approaches to implementing these methods still have important limitations, however. We demonstrate three such problems: errors due to the mixture of image properties across boundaries, an intrinsic ambiguity in boundary location when only short sequences are considered, and difficulties insuring that the motion of a boundary aids in flow estimation for the surface to which it is attached without corrupting the flow estimates for the occluded surface on the other side. The first problem can be fixed by basing flow estimation only on image changes at edges. The second requires an analysis of longer time intervals. The third can be aided by using a boundary detection mechanism which classify the sides of boundaries as occluding and occluded at the same time as the boundaries are detected.

1 Introduction

Discontinuities in optical flow are normally viewed as a serious impediment to producing accurate estimates of the flow. Due to the ambiguous nature of flow within small space-time neighborhoods of an image sequence, all methods for estimating flow either explicitly or implicitly assume some sort on spatial and/or temporal continuity. This assumption is often violated at surface boundaries. This paper points out important shortcomings in one class of methods for estimating possibly discontinuous optical flow and indicates how improvements can be made.

Two general approaches to dealing with flow discontinuities are found in the literature. The first explicitly allows for a mixed distribution near boundaries. Marr and Poggio (1979) and Barnard and Thompson (1980) did this by basing their analyses on prominent modes of possible disparities over local neighborhoods. Using currently popular terminology, the modes acted as a *robust estimator* for the dominant disparity in the neighborhood, though neither paper explained this effect. Some years later, Prazdny (1985) made the idea explicit. Scott's principal component approach (Scott 1988) and Schunck's constraint line clustering (Schunck 1989) utilize the same concept. More recently, Black and Anadan (1990) explicitly used robust estimation in determining optical flow near boundaries. Bergen et al. (1990) used a more specialized approach

This work was supported by National Science Foundation grant IRI-9112267.

in which one dominant motion is found first and then used to produce a new image sequence in which the found motion is nulled, allowing a second motion to be more easily estimated. Once flow is estimated using any of the above approaches, it is relatively straightforward to find the flow discontinuities (Thompson et al. 1985).

The second approach is to make discontinuities explicit, thus avoiding the need to deal with mixed distributions. All of these methods follow the basic structure of Geman and Geman (1984) or Blake and Zisserman (1987). The most common approach utilizes a Markov random field (MRF) formulation with explicit line processes (Murray and Buxton 1987; Gamble and Poggio 1987; Hutchinson et al. 1988; Koch et al. 1989; Konrad and Dubois 1992; Heitz and Bouthemy 1993). Interacting region and line processes are involved. The region processes combine information over local neighborhoods to reduce ambiguity and improve reliability of individual flow estimates. The line processes estimate locations of discontinuities and act to keep points on opposite sides from interacting in the region processes. The estimation of discontinuities can be greatly assisted if independent information about possible surface boundaries is available. In particular, most of the papers referenced above allow the assertion of a boundary element only at locations likely to correspond to contrast edges.

This paper addresses optical flow estimation using explicit discontinuity detection. We provide evidence for three claims:

- Line processes, as usually implemented, do not completely remove undesirable interactions between surfaces on either side of a motion boundary.
- There is an intrinsic ambiguity in the localization of motion boundaries which can only be resolved over multiple frame pairs.
- Distinguishing between the occlud*ing* and occlud*ed* sides of an optical flow boundary can aid the flow estimation process.

2 Estimating Optical Flow in the Presence of Discontinuities

Existing flow estimation techniques which involve explicit motion boundary detection suffer from two deficiencies when applied to image sequences with textured backgrounds. Interactions between surfaces on either side of the boundary can still occur, degrading the accuracy of the estimated flow. Furthermore, motion boundary detection itself is subject to a localization ambiguity when only a single frame pair is analyzed at a time.

2.1 Mixed image properties near the boundary

Clearly, optical flow cannot be determined based only on a single space/time image point. All methods for estimating optical flow use image properties that must be computed over neighborhoods. Gradient-based techniques (e.g, Fennema and Thompson 1979; Horn and Schunck 1981) require the estimation of spatial and temporal derivatives using some form of finite difference. Region-based methods (e.g., Anandan 1989) compare local neighborhoods around selected points in each frame. Space-time filtering methods (e.g., Adelson and Bergen 1986; Heeger 1988; Fleet and Jepson 1990) are implemented using FIR filters with finite spatial and temporal extent. Image properties are often computed after some form of pre-filtering (Kearney et al. 1987; Barron et al. 1994), also involving operations with finite spatial and/or temporal extent. In addition, many flow estimation methods impose explicit flow continuity constraints (e.g., Fennema and Thompson 1979; Barnard and Thompson 1980; Horn and Schunck 1981; and the various MRF algorithms).

The line processes used to deal with flow discontinuities in MRF-like methods in fact only affect the imposition of explicit flow continuity constraints. Image properties associated with points to one side of an asserted boundary can still be effected by the intensity patterns on the other side of the boundary. Derivatives estimated using finite differences and pre-filtering will have substantial errors in the vicinity of motion boundaries, particularly if the surfaces to one of both sides are highly textured. A similar effect occurs with spatiotemporal filters. While the line processes stop direct interactions between estimated flow to either side of a surface boundary, the flow estimates themselves can still be highly inaccurate.

Mixed image properties at boundaries are one of the reasons for using matching methods based on multiple window sizes, though this rationale is seldom stated explicitly. Perhaps this is the reason that few such methods systematically deal with the problem. (Kanade and Okutomi (1994) is one exception.)

Only methods which are based on correctly chosen sparse image features are immune from this effect. The features must be lines or points that necessarily occur either well away from surface boundaries or occur exactly at the boundary and for which localization is not significantly effected by image structure to either side of the boundary. The Moravec "interest operator" (Barnard and Thompson 1980), extrema in the difference of Gaussians (Mutch and Thompson 1983), corner detectors (Rangarajan et al. 1988), and methods based on contrast edges (Hildreth 1983; Waxman et al. 1988) satisfy this property.

2.2 Ambiguity of surface boundary location

Many of the methods for detecting flow discontinuities are based on looking for large magnitude values of the flow gradient. Unless the location of the flow discontinuity is already known or methods which allow for multiple motions near boundaries are used, estimated flow will be smoothed over near the boundary. Figure 1 illustrates the problem. Estimated flow magnitude is plotted against position along an axis perpendicular to a surface boundary. Away from the boundary, flow estimates will be reasonably accurate. They should also be accurate on the occluding surface near the boundary, since the boundary itself will have sufficient structure to allow flow estimation and is moving with the occluding surface. On the occluded side, however, flow estimates won't be accurate until observed sufficiently far from the boundary to allow the moving texture of the occluded surface to dominate any effects associated with the surface boundary, which is moving with a different optical flow. The maximum rate of change in flow will typically be noticeably offset from the actual boundary.[1]

Whatever flow-based boundary detection method is used, there is an intrinsic localization ambiguity associated with instantaneous flow that cannot be overcome without either non-motion cues to the boundary location or analysis over longer time intervals (Thompson and Barnard 1981).[2] Consider the pattern shown

[1]The optical flow edge detector described in (Thompson et al. 1985) used a robust flow estimation process that did *not* smear flow values near boundaries. It was still subject to the problems described in the next paragraph, however.

[2]There is evidence that the human vision system resolves this ambiguity by favoring whichever possible boundary has the strongest non-motion edge properties (Yonas et al. 1990).

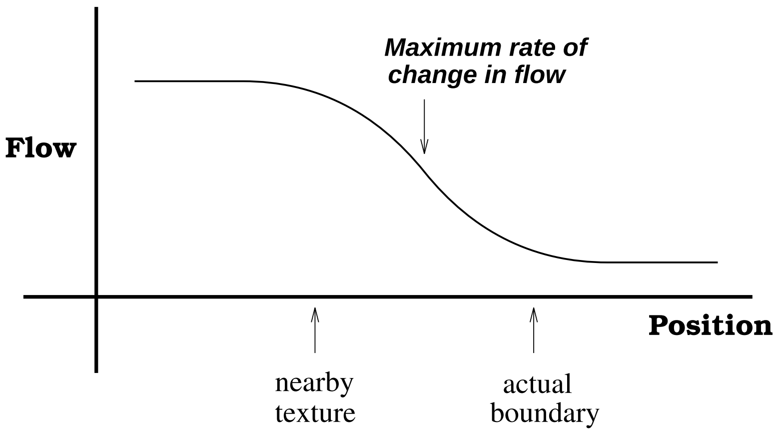


Figure 1: Optical flow gradients are not sufficient to accurately locate flow boundaries.

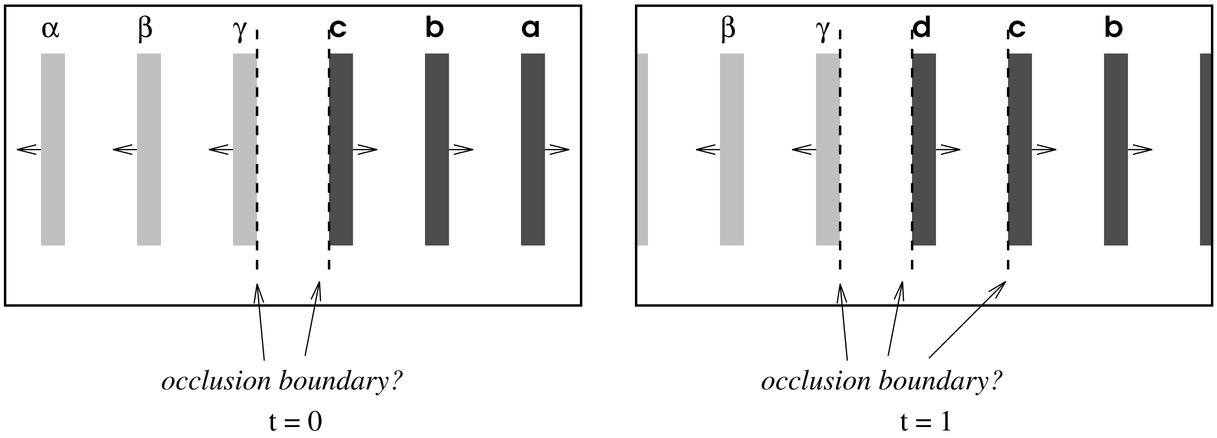


Figure 2: Intrinsic ambiguity in boundary location at one instant in time.

on the left in Figure 2. Three bars, α, β, and γ, are moving to the left. Three other bars, **a**, **b**, and **c**, are moving to the right. Assuming that occlusion boundaries are likely to have associated contrast edges, then the edge of the occluding surface is either along the right side of bar γ or the left side of bar **c**. Without additional information, there is no way to resolve this ambiguity. Not until the next time step, shown on the right in the figure, is it apparent that the lighter colored bars on the left are occluding the darker colored bars to the right and that the true occlusion boundary is at the edge of bar γ. This problem affects not only boundary detection based on flow gradients, but also methods which are based on the appearance and disappearance of surface texture suffer from a related effect (Kaplan 1969; Mutch and Thompson 1985), methods which allow for multiple motions within a local neighborhood, and methods which attempt to recognize occlusion boundaries using properties of similarity surfaces (e.g., Anandan 1989). A method for deciding which possibility – bar γ, bar **c**, or the newly appeared bar **d** – really corresponds to the surface boundary is described in section 3.2.1.

3 An Algorithm for Flow Estimation in the Presence of Discontinuities

This section describes an optical flow estimation algorithm which effectively addresses the problems discussed in section 2. It has similarities to the method presented in (Heitz and Bouthemy 1993), though differs in a number of important regards: the algorithm correctly deals with mixed image properties near surface boundaries, it uses a more complete classification of boundary types, and it is able to resolve boundary ambiguities when textured backgrounds are present. The algorithm utilizes explicit line processes to avoid undesirable interactions across motion discontinuities in a manner analogous to the MRF methods, but using a much simpler formulation. In the remainder of this paper, our analysis excludes motions consisting exclusively of rotations of revolute objects around an axis perpendicular to the line of sight and corresponding to the axis of symmetry of the object. This particular situation introduces additional ambiguities, the resolution of which is still largely an open question (Thompson et al. 1992; Thompson and Painter 1992)

3.1 Dealing with discontinuities

At flow discontinuities, effective flow estimation methods must avoid smoothing the flow across the boundary and must also keep image properties associated with a surface on one side of the boundary from corrupting the flow estimation for the surface on the other side. We deal with this second problem by basing flow estimation solely on image features which are relatively unaffected by this mixture of image properties. In particular, we base flow calculations only on image properties at edges. As long as no temporal pre-filtering is performed, spatial contrast edges will either be close approximations to surface boundaries or far enough away from surface boundaries so that local image properties near the edge are largely unaffected by image regions that are part of other surfaces. This has the additional advantage of minimizing difficulties due to apparent misregistrations between contrast and motion edges due to localization errors when separate image primitives are used. The standard Horn and Schunck (1981) method is utilized, modified so that spatial and temporal image gradients are only used at spatial edges. For other image locations, flow is estimated using the interpolation properties intrinsic to the method.

Given information about the location of surface boundaries, it is straightforward to avoid flow computation interactions across boundaries. The Gauss-Seidel method used in the basic Horn and Schunck algorithm bases flow estimates for a given pixel at a particular iteration step on the average values at neighboring pixels in the previous iteration. It is only necessary to make sure this average does not include any pixels on the other side of a motion discontinuity.

Discontinuity detection is based on an analysis of flow differences across contrast edges which potentially signal surface boundaries. When possible, the surfaces to either side of a detected boundary are classified as occlud*ing* or occlud*ed* using the *boundary flow constraint*, which states that the flow associated with the occluding surface immediately adjacent to the boundary will be equal to the flow of the boundary (Thompson et al. 1985). Violations of the boundary flow constraint can be used to identify occluded or disoccluded surfaces, as shown in Figure 3, without any need to know the camera or object motions involved. (When rotation is depth is possible, identification of occlud*ing* surfaces is more complex.) This relationship is only useful when there are differences across the boundary in the component of flow normal to the boundary. If the flows to either side of a boundary are parallel to the boundary itself, then it is not possible to determine which side is occluding the other without additional information about camera motion. Such *sheer boundaries* still indicate occlusion, however.

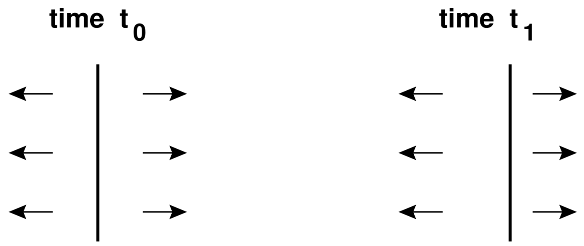


Figure 3: The surface to the left has an optical flow different from that of the boundary and hence is an occlud*ed* surface. Since it is moving away from the boundary, it is in fact being *dis*occlud*ed*.

To detect and classify motion boundaries, the flow associated with every point on a contrast edge is compared to the flow to either side. Only the flow, $f^{\perp}$, of the edge normal to its orientation is considered. The flows to either side are separated into components normal to the edge orientation, $f_1^{\perp}$ and $f_2^{\perp}$ and parallel to the edge, $f_1^{||}$ and $f_2^{||}$. Detection and classification proceeds by first checking to see if one surface is progressively occluding or disoccluding the other. If not, sheering surfaces are checked for:

$d_1^{\perp} = | f_1^{\perp} - f^{\perp} |$, $d_2^{\perp} = | f_2^{\perp} - f^{\perp} |$
if $\max(d_1^{\perp}, d_2^{\perp}) \geq T_1$
    if $d_1^{\perp} > d_2^{\perp}$
        side 1 is *occluded*, side 2 is *occluding*
    else
        side 1 is *occluding*, side 2 is *occluded*.
else if $| f_1^{||} - f_2^{||} | \geq T_2$
    sides 1 and 2 are *sheer*

This method, while less elegant that the detection and classification scheme described in (Thompson et al. 1985), is more reliable. Note in particular that $f_1^{\perp}$ and $f_2^{\perp}$ are separately compared to the normal flow of the edge, rather than just taking their difference. This is important, as it increases the sensitivity of the edge detection. Because of the effects described in section 2.2, the difference in flow normal to the edge attributable to one surface moving relative to another will be concentrated in $d_1^{\perp}$ or $d_2^{\perp}$. Additional differences in flow normal to the edge will arise due to smooth variations in flow on the occluding side of the edge and from various noise effects. As a result, basing detection on $| f_1^{\perp} - f_2^{\perp} |$ will increase the classification error rate.

Flow estimation at a boundary pixel neither affect nor are affected by the flow at pixels to the occluding side of the boundary. Interactions to the occluding side, however, are allowed, since the boundary is part of that surface. A conservative approach is taken for sheer boundaries. The flow estimation at the boundary is isolated from the flow values to either side, but does interact with the estimation of other values along the boundary. The complete algorithm interleaves flow estimation and boundary detection. Enough Gauss-Seidel iterations are done to get a reasonable estimate of flow values. Boundaries are then detected and classified. Better flow estimates are obtained by performing additional iterations using this information. The process is repeated as necessary.

3.2 Improving boundary localization over time

Optical flow estimation over sequences longer than a single frame pair can improve the efficiency of iterative algorithms by providing reasonably accurate initial values and can improve the accuracy of many methods by using some form of temporal coherence to reduce the effects of noise. It turn out that longer sequences are also the key to resolving the localization ambiguity described in section 2.2.

3.2.1 Boundary projection

Surface boundaries persist over time. Once these boundaries are found at one time step, their position can be easily predicted for the next time step. This is done by taking each point on a detected occlusion boundary and looking for a contrast edge in the future frame at the corresponding location, offset by the flow of the boundary point. Substantial efficiency is achieved by starting the iterations at the next time step using these predicted boundary locations. Even more importantly, over time actual surface boundaries will be maintained while detected boundaries that in fact correspond to surface texture near true boundaries will disappear.

To see why this is so, consider Figure 2. At time step 0, the interleaved iterate and classify algorithm will end up finding potential occlusion boundaries to the right of bar γ and to the left of bar **c**. The flow estimates assigned to the region between the bars is likely to be a muddle. At time step 1, bar **d** has appeared. The flow of this new "texture element" will initially propagate both left and right, since there is no surface boundary indication associated with the bar. As the flow propagation approaches bar **c**, the differential flow across the (false) boundary on the left side of **c** is reduced, leading to a reclassification of the edge as non-occluding. The sides of **d** never get classified as occluding, since the only flow not consistent with the edge motion is blocked by the boundary at γ from propagating towards **d**. The situation shown in Figure 2 is extreme in that a full texture element appears over a single time step. In practice, texture element spacing is usually much larger than inter-frame motions and the effect described above takes place more gradually.

3.2.2 Flow projection

Horn and Schunck suggest improving the efficiency of their algorithm by using the results obtained for one time step to initialize the iterations at the next time step. Simply using the flow obtained for each pixel at one time step to initialize the estimates for the same pixel at the next time step is adequate over smooth surfaces, but can actually lead to worse results at occlusion boundaries than if a default of zero flow is used in the initialization. This is because the flow associated with a disoccluded surface region will typically be very different from the flow of the surface which was previously occluding it. Some improvement can be obtained by projecting the flow estimates at one time step to pixels in the next time in a manner that takes into account the flow value itself. This fails, however, to provide initial estimates for flow in occluded regions where multiple flow values project to the same point and in disocclusion regions where no flow values from the previous time step will project (Black and Anandan 1990).

Effective projection of flow into occluded and disoccluded regions in the next time step is possible if occlusion boundaries have been detected and classified in the current time step. Figure 4 shows a simplified situation in which an occlusion boundary is stationary, the occluding surface is to the left, the occluded sur-

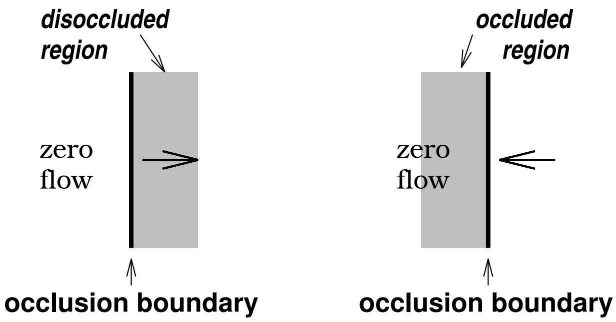


Figure 4: Projecting flow into occlusion and disocclusion regions.

face is to the right, and the motion of the occluded surface is normal to the boundary. Two possibilities exist: the occluded surface is moving either towards or away from the boundary. (In the case of pure sheer motion, no occlusion or disocclusion regions exist.) Movement away from the boundary causes disocclusion. The gray area on the left of Figure 4 indicates the region in the next time step that will correspond to surface visible for the first time and which will have flow close to that on the occluded side of the boundary. Movement towards the boundary causes occlusion. The gray area on the right in Figure 4 indicates the region in the next time step where flow vectors from two different surfaces will both project. The best estimate of the actual flow will be the flow of the boundary itself. In either case, the region of occlusion/disocclusion runs from the boundary to a line found by adding the flow vector of the occluded surface point nearest each boundary point to the boundary point location. For the more general case of moving boundaries, the region of occlusion/disocclusion runs from the projection of the boundary into the next time step to the line found by adding the flow vector of the occluded surface point nearest each boundary point to the projected boundary point location. The flow to be filled into these regions is either the flow of the boundary or the flow of the nearest occluded surface point, depending on the classification of the boundary.

Note that the principal goal here is to improve efficiency, not implement some sort of temporal continuity constraint. Temporal consistency can also be used to improve accuracy in flow estimation, though this constraint can lead to problems in occlusion/disocclusion regions (Murray and Buxton 1987). Thus, the approach described in the preceding paragraph might be extended to better exploit temporal continuity at and near boundaries.

3.3 Implementation details

It is important the the spatial derivatives, temporal derivatives, and contrast edges used in our flow estimation algorithm all refer to the same set of points. We accomplish this by using NTSC-style interlaced imagery. Image sequences are a sequence of field pairs, with each field offset a half inter-frame time and one inter-pixel distance in the vertical direction. To maintain comparable horizontal and vertical resolution, we subsample every other pixel on each line, starting at the first pixel for even fields and the second pixel for odd fields. Spatial and temporal derivatives are computed using the 2x2x2 cube of (Horn and Schunck 1981), applied to the even fields of successive frames. Edge detection is done on the intervening odd field, in a manner that assigns the edge location to the nearest pixel. The result is that both derivatives and edges correspond to integral pixel locations in the middle field.

The Canny edge operator was used in all of the experiments described below, though a zero-crossing edge detector gives comparable results. No thresholding based on edge strength was used. Gaussian pre-smoothing using a standard deviation of 1.0 was done prior to both edge detection and space/time derivative estimation. Unless otherwise stated, a total of 150 iterations were performed at each time step. When boundary classification was performed, it was done after 1/3rd, 2/3rds, and the full set of iterations. Flow values 2 pixels to either side of contrast elements were used for boundary classification. $\alpha = 2.0$ was used in the Horn and Schunck update procedure.

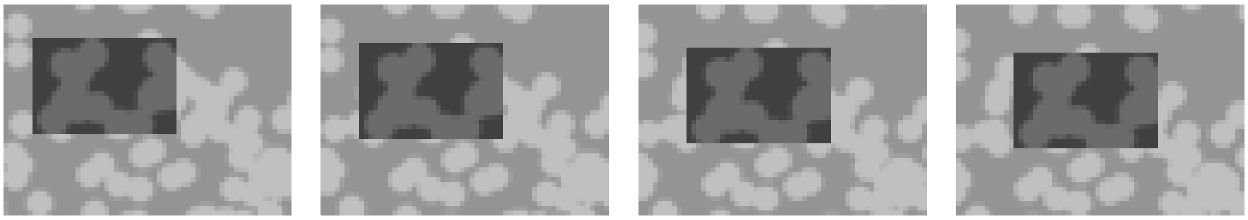
4 Experimental Results

Most of the experiments described in this section were done using synthetic data. While the limitations of using synthetic data are well known, it is the only way to do fair quantitative comparisons between alternate approaches. The accuracy of estimated flow was measured using the inter-frame angular deviation described in (Barron et al. 1994). While this measure is useful in evaluating the results of the controlled experiments described below, the absolute magnitudes of flow errors should not be compared with those obtained in other circumstances. In the tests reported here, most of the flow estimation error occurs at flow discontinuities, which correspond to a substantial number of the pixels in the test sequences. This is not the case with other image sequences, such as those used in (Barron et al. 1994). Error magnitude comparisons between dissimilar image sequences will be dominated by the number of pixels at or near discontinuities at least as much as by the quality of the flow estimation algorithms applied.

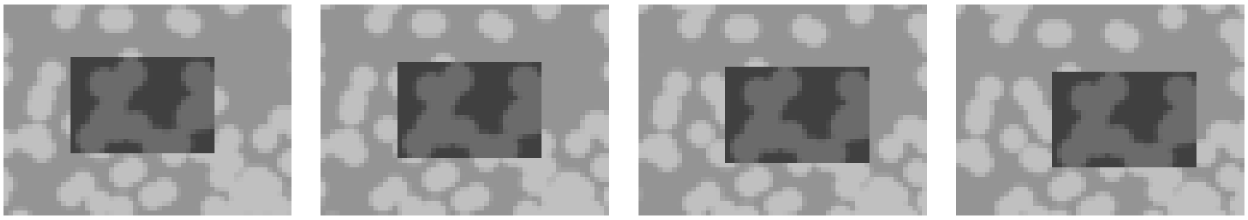
In evaluating the results, it is also important to note that the performance of the methods described here depends on both the particular motion analysis algorithm used and the quality of the (static) edges on which it is based. In order to provide a grounds for comparing algorithms, a simple edge detector was used and manipulations of the test sequences were structured to avoid perturbations of edge detector performance. For example, sub-pixel motions were avoided, since edge detector results would be affected by the anti-aliasing scheme used to generate synthetic images with non-integral motions. To get the best possible performance, as would be desirable if quantitative comparisons were been done with competing methods, a more sophisticated edge detector should be used.

Figure 5 shows the first eight intermediate (odd field) images from an 120 by 160 pixel synthetically generated sequence with a textured rectangle moving across a textured background. The rectangle is moving four pixels to the right and two pixel down per frame. The background is moving two pixel down and to the left per frame. Due to the subsampling described in Section 3.3, the images that were processed were 60 by 80, with the effective translation between frames half that described above. Figure 6 shows the true occluded/occluding and sheer boundaries for this sequence. In this and subsequent figures, sheer boundaries are marked by a light gray pattern on either side of a dark line while occluded/occluding boundaries are marked by a darker gray pattern on the occluded side of the edge.

Figure 7 shows the optical flow estimated by the standard Horn and Schunck algorithm, applied to the first four frame pairs of the moving rectangle sequence. The errors in flow estimation, particularly at and near the boundaries of the moving rectangle, are obvious. Figure 8 demonstrates the results of adding explicit boundary analysis to the basic Horn and Schunck algorithm. The final flow estimates and boundary



Frames 1 – 4



Frames 5 – 8

Figure 5: Original image sequence.

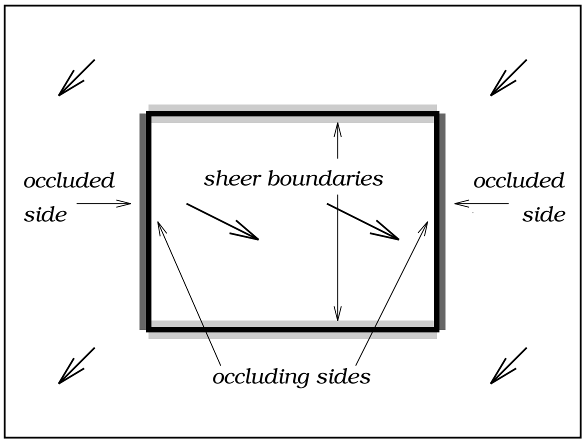


Figure 6: “True” boundary classifications for moving rectangle test sequence.

classifications at each time step are shown. While the flow estimates are improved, substantial error remains due to the mixing of image properties across the boundaries, which seriously distorts the gradient computations. Figure 9 uses the same algorithm as Figure 8, except that only image properties at contrast edges are used in the flow computations. This minimizes the effects of image property mixing between surfaces. Results for eight frame pairs are shown. Flow estimates are further improved. The ambiguity in detecting and classifying boundaries when background texture appears near occluding contours, which is manifested in the figure as missing and extraneous boundary segments, is still affecting accuracy, however. Figure 10 shows the effect of using boundary projection to initialize processing at the next time step. For each frame pair, the estimated flow, estimated boundaries based on the previous frame pair, and detected boundaries in the current pair are shown one on top of the other. The boundary classification errors have largely been

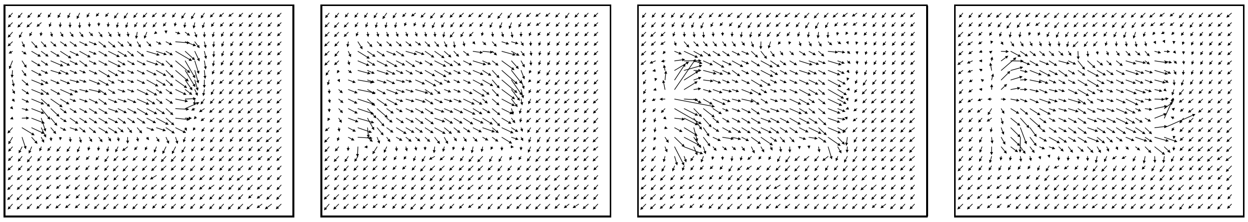


Figure 7: Flow estimation using standard Horn and Schunck method (first four frame pairs).

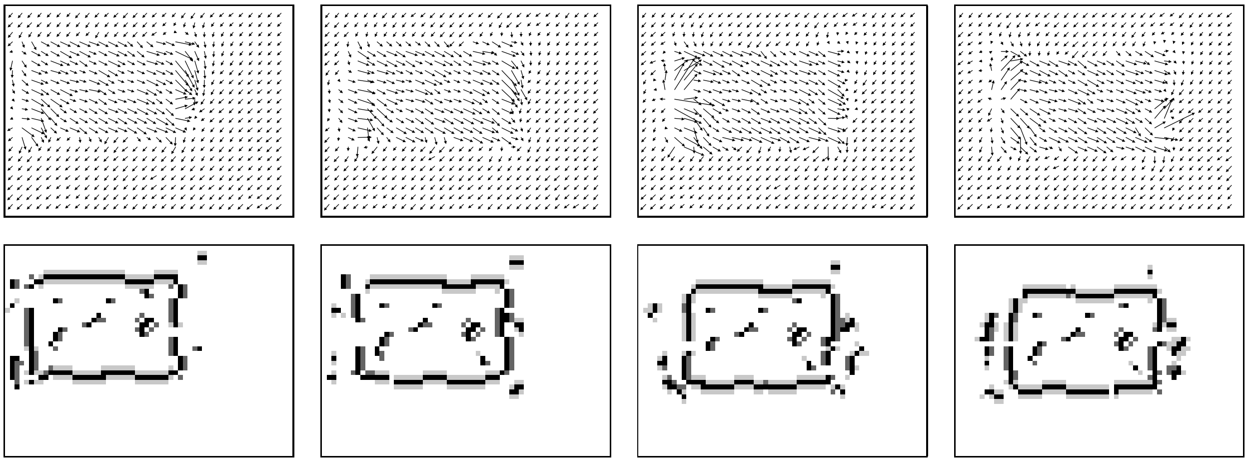
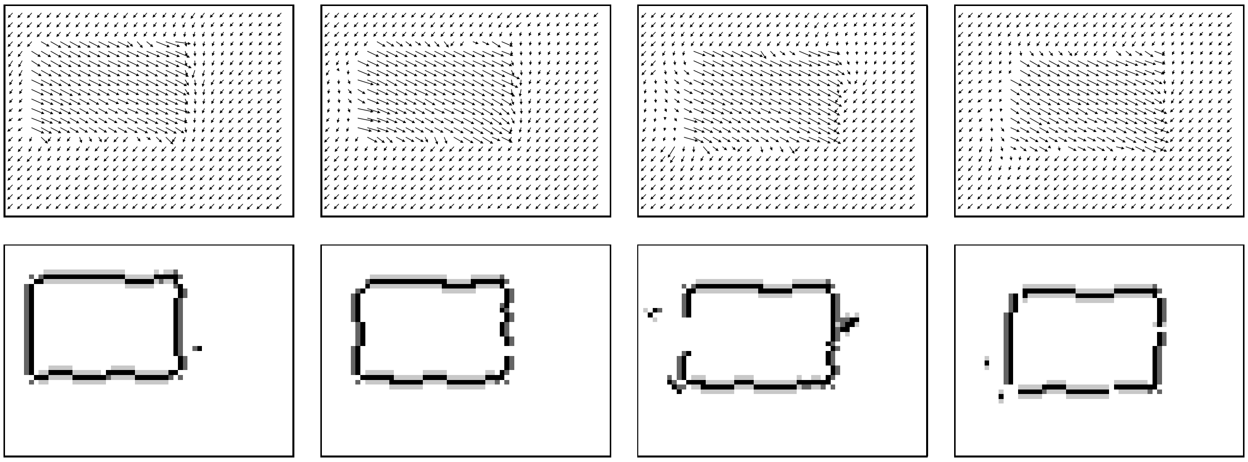


Figure 8: Flow estimation using Horn and Schunck with explicit boundary detection (first four frame pairs).

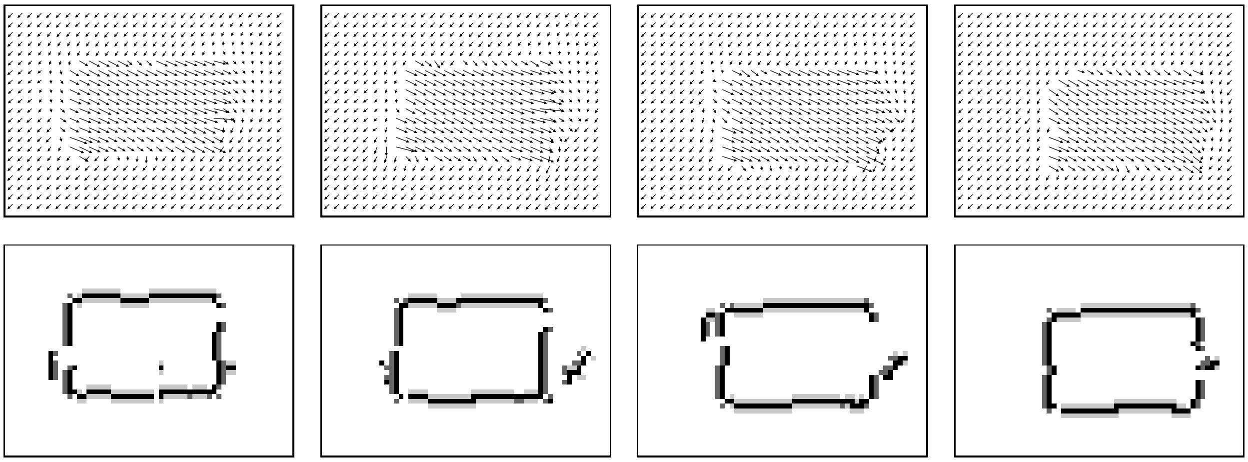
eliminated and the flow estimates correspondingly improved. Table 1 lists the quantitative average errors associated with each of these algorithms.

The next set of experiments test the value of classifying surface boundaries by labeling the sides as occluding or occluded. A test sequence consisting of a uniform intensity disk moving over a textured background was used, with foreground and background motions as in the previous tests (Figure 11). To insure convergence given the large, uniform brightness area, 600 iterations were run at every time step. Occluding/occluded boundary classification should allow the motion of the disk boundary itself to propagate into the otherwise featureless interior of the disk. Figure 12 shows the flow estimate when detected discontinuities are used to stop smoothing across the boundary but otherwise are not allowed to affect flow computations. This is a situation that can commonly arise in MRF formulations of flow estimation. Its seriousness depends on implementation details of such methods. Figure 13 shows the flow estimated using full boundary classification. As is clear in Table 2, accuracy is improved.

The final set of experiments on synthetic data address the issue of how well projecting the estimated flow from one time step to initialize computations at the next time step helps in flow determination. The image sequence used was that shown in Figure 5. A total of 30 iterations were run at each time step, as compared to 150/step for the results shown in Figures 7-10. Five cases were examined, with the resulting errors shown in Table 3. For the first case, the algorithm employing boundary classification and boundary projection was used as in Figure 10, only with the reduced number of iterations. In the second case, no boundary projection

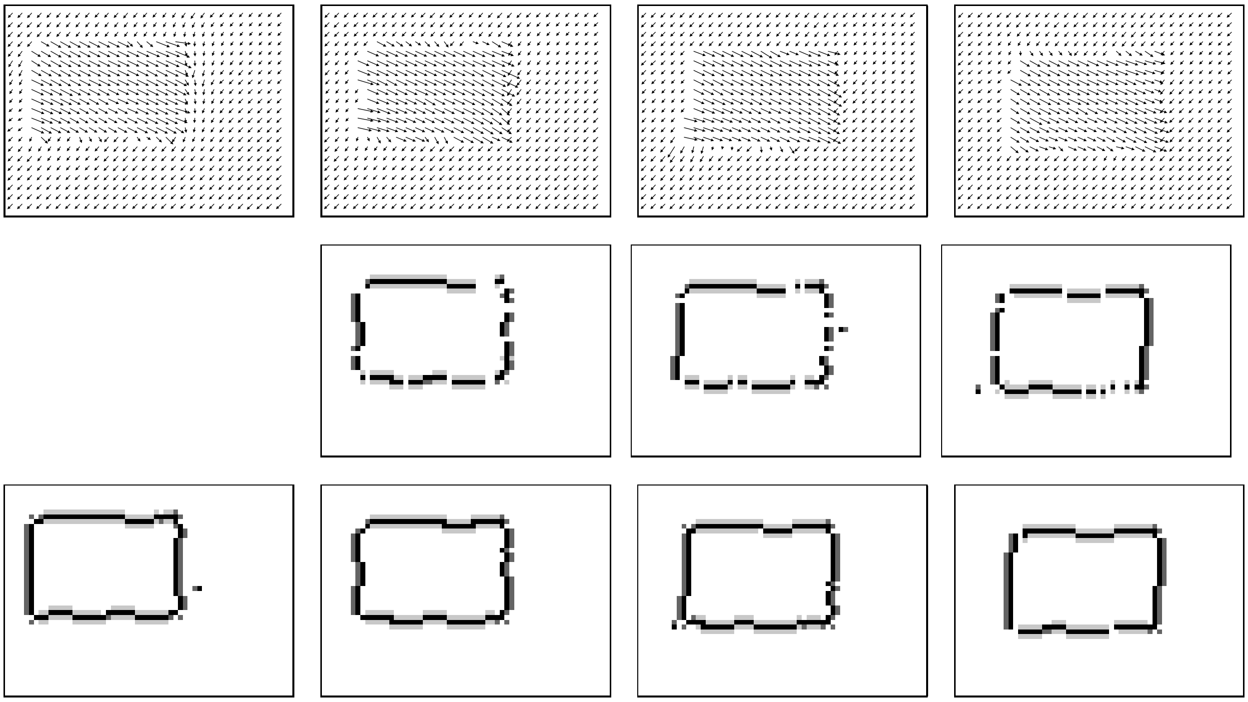


Frames 1 – 4

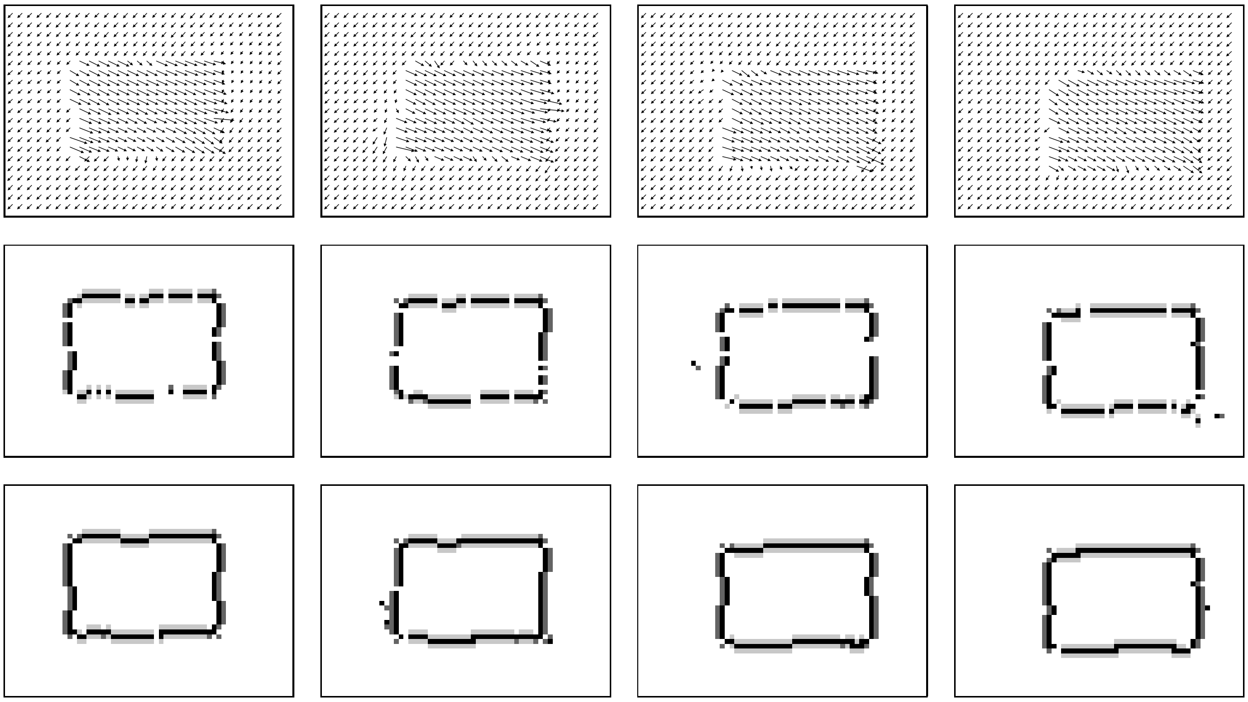


Frames 5 – 8

Figure 9: Edge-based flow estimation using explicit discontinuity detection.



Frames 1 – 4



Frames 5 – 8

Figure 10: Flow estimation using explicit discontinuity detection and projection.

frame pair:	1	2	3	4	5	6	7	8
Basic Horn and Schunck	13.38°	14.05°	15.91°	15.62°	13.14°	15.52°	13.84°	13.43°
Horn and Schunck with boundary detection	11.31°	11.75°	13.27°	12.88°	11.18°	13.90°	11.77°	10.88°
Edge-based flow estimation with boundary detection	7.55°	8.28°	8.76°	7.15°	10.38°	9.74°	9.02°	7.09°
Edge-based flow est. with boundary detection, projection	7.55°	5.35°	4.75°	4.65°	5.28°	5.29°	4.81°	4.25°

Table 1: Flow errors for different estimation algorithms.

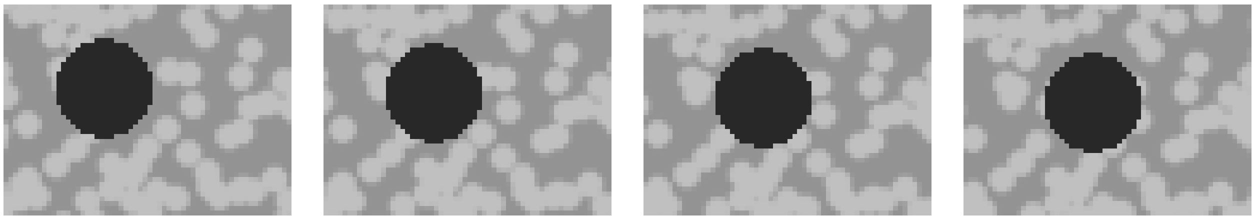


Figure 11: Moving disk image sequence.

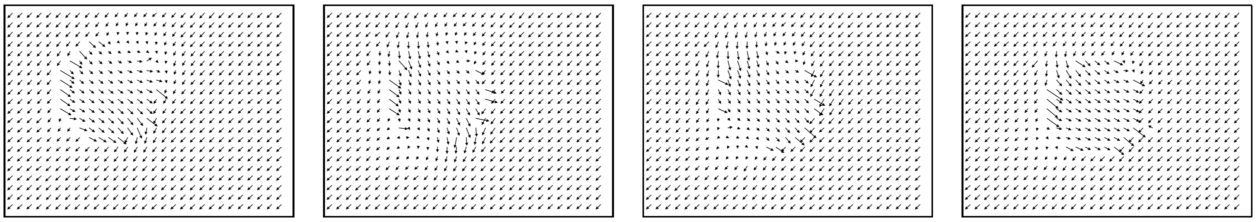


Figure 12: Flow estimation using symmetric boundaries.

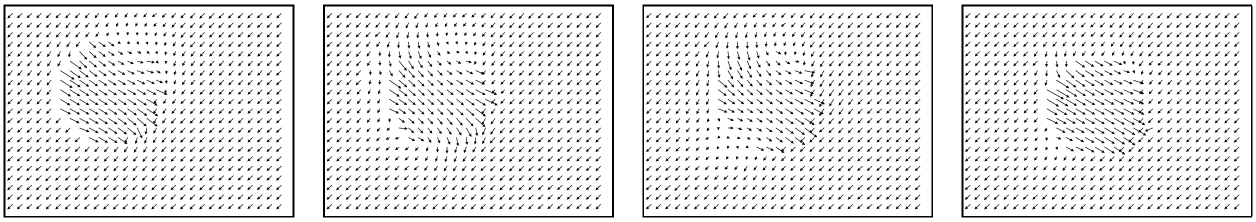


Figure 13: Flow estimation using asymmetric boundaries.

frame pair:	1	2	3	4	5	6	7	8
Symmetric boundary classification	6.75°	9.32°	8.51°	5.28°	5.40°	6.62°	6.81°	7.07°
Asymmetric boundary classification	5.55°	7.26°	7.00°	3.86°	4.07°	4.67°	4.95°	5.50°

Table 2: Flow errors using symmetric and asymmetric boundary classification.

frame pair:	1	2	3	4	5	6	7	8
Boundary projection, no flow projection	21.99°	22.95°	22.37°	23.28°	22.74°	23.24°	22.65°	21.15°
No boundary projection, copied flow	21.99°	14.93°	13.46°	12.53°	12.43°	12.66°	12.25°	12.26°
No boundary projection, projected flow	21.99°	15.88°	13.34°	11.89°	11.29°	11.06°	10.96°	9.96°
Boundary projection, copied flow.	21.99°	14.47°	11.59°	10.23°	10.07°	9.50°	9.06°	9.39°
Boundary projection, projected flow.	21.99°	15.01°	10.86°	8.74°	7.58°	7.04°	6.67°	5.56°

Table 3: Flow errors when total iterations are reduced.

was done, but flow in each time step after the first was initialized copying the values obtained in the previous time step. (While boundaries were not projected from frame to frame, boundary detection and classification was used within each time step.) With reduced iteration, initializing flow estimates is clearly more valuable than initializing boundary estimates. In the third test, there was no boundary projection and flow estimates were initialized using the projection method described in section 3.2.2. There is a substantial improvement in accuracy. The last two cases use boundary projection and flow that is initialized by either copying or projecting values from the previous time step. Again, flow projection is clearly superior to simple copying.

Figure 14 shows the results of using explicit boundary detection and projection on a real image sequence. The method used was the same as for calculating the results shown in Figure 10. The camera is tracking a person walking to the right. As a result, in the image the dominant motion is the background moving to the left. Since the tracking is imperfect, there is also residual image motion of the person moving to the right. Though the boundary detection and classification is not perfect, it does a reasonable job of distinguishing between occluding and occluded surfaces, even though the magnitude of the image motion of the foreground object is less than the magnitude of the background motion. The irregularity of the detected motion boundaries is due directly to gaps and errors in the Canny edge detector, since the method as stated only looks for motion boundaries where contrast edges have already been found. This suggests that for realistic scenes in which objects of interest do not have high contrast boundaries with respect to the background, that the all-or-nothing approach of only allowing the assertion of boundaries at intensity edges should probably be relaxed.

Figure 15 shows a wind-up toy moving to the left, imaged with a camera mounted on a tripod. There is little brightness variation across most of the object, a situation which is quite difficult for most gradient-based flow estimation algorithms to deal with. The figure clearly shows how the effectiveness of the method increases over multiple frames, with appropriate flow assigned to the moving object only after boundaries have been determined.

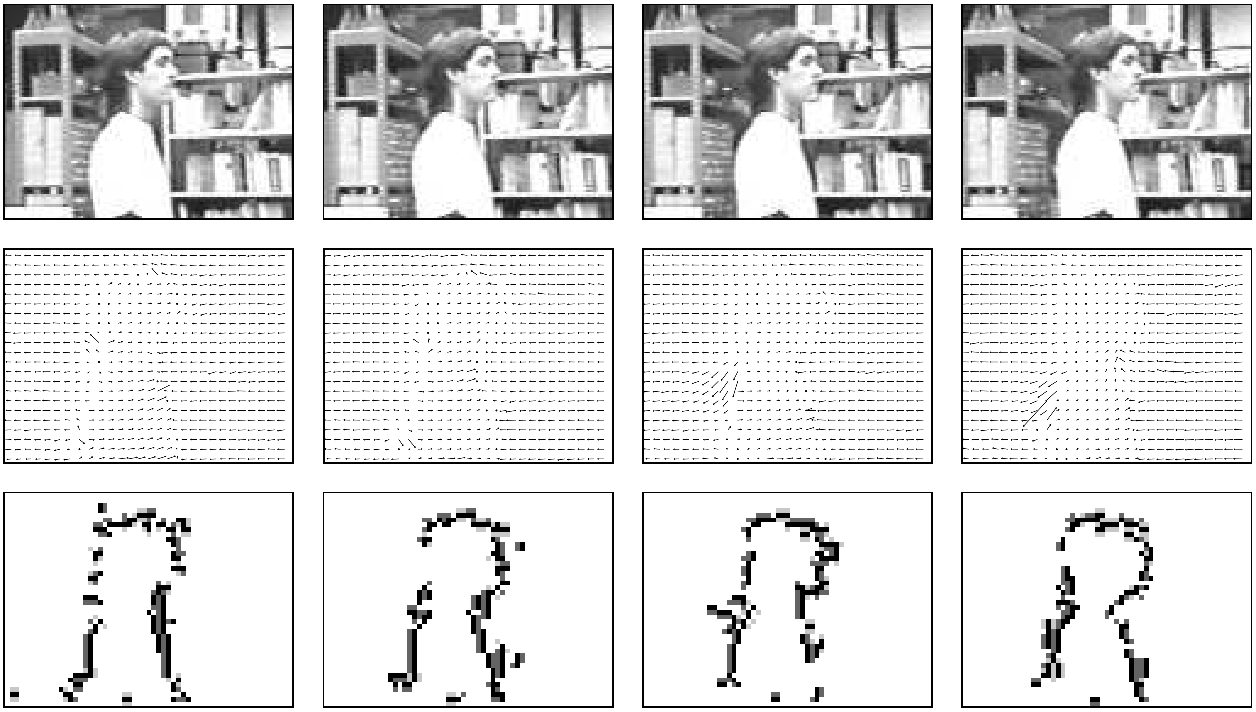


Figure 14: Image sequence, estimated flow, boundary classifications: camera tracking object.

5 Summary

This paper has examined a number of important issues affecting the accuracy of optical flow estimates made by algorithms which explicitly take into account flow discontinuities. Theoretical and experimental evidence has been presented showing that the line processes typically used to keep flow smoothness constraints from corrupting estimates across boundaries can still allow image properties associated with a surface to one side of a boundary to improperly influence flow computations on the other side of the boundary. A second problem affecting flow estimation is the intrinsic ambiguity of boundary localization in textured environments when only two frames are considered at a time. This effect was clearly demonstrated using one particular flow estimation algorithm, but occurs independent of the algorithm used. Experimentation has shown the value of distinguishing between occluding and occluded sides of a surface corresponding to a flow discontinuity. This information can be used to improve flow estimates and is important in projecting flow estimates to future time steps so as to reduce computational requirements. Finally, we have shown how the line processes used in MRF formulations for estimating optical flow can also be used in the conceptually simpler Horn and Schunck algorithm. While the techniques we have described for improving flow estimation at boundaries are most easily implemented in algorithms such as Horn and Schunck and MRF methods, they can be applied across the spectrum of approaches to optical flow estimation. For example, the area correlation algorithm described in (Smitley and Bajcsy 1984) can be viewed as utilizing something akin to a line process when matching image regions.

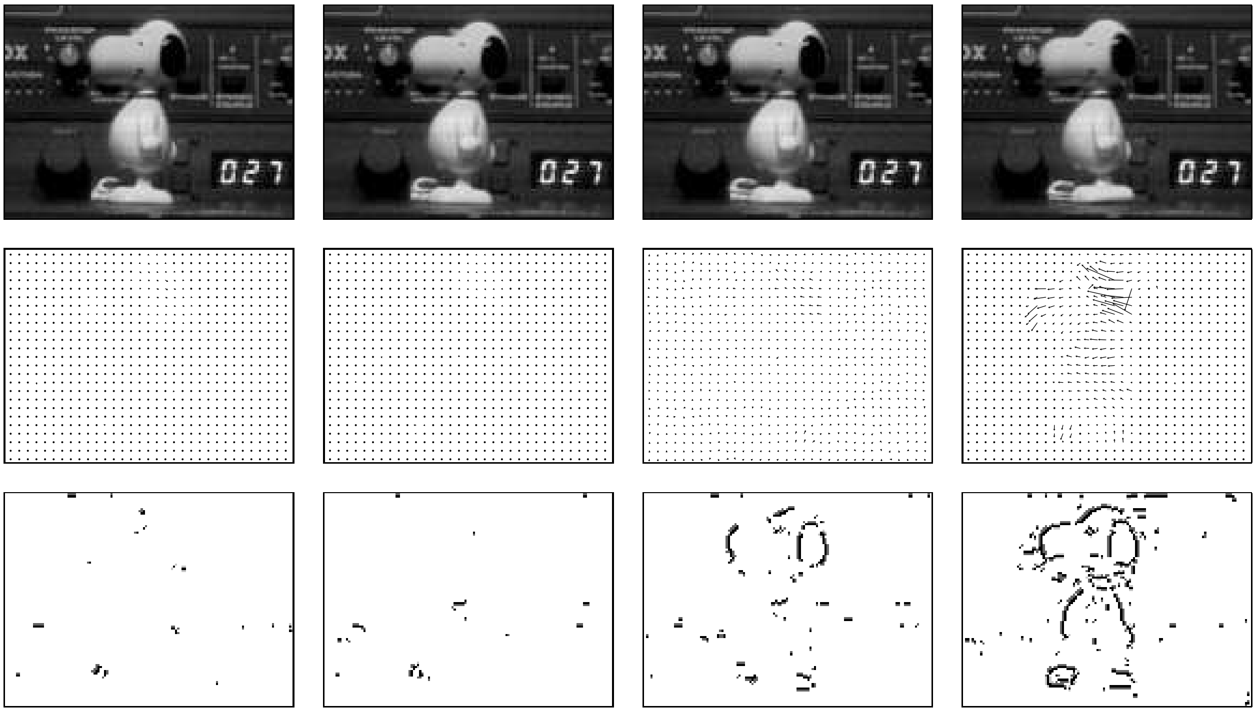


Figure 15: Image sequence, estimated flow, boundary classifications: moving object with near-homogeneous surface.

References

Adelson, E. H. and J. R. Bergen (1986, May). The extracton of spatio-temporal energy in human and machine vision. In *Proc. Workshop on Motion: Representation and Analysis*, pp. 151–155.

Anandan, P. (1989, January). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 283–310.

Barnard, S. T. and W. B. Thompson (1980, July). Disparity analysis of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-2*, 333–340.

Barron, J. L., D. J. Fleet, and S. S. Beauchemin (1994, February). Performance of optical flow techniques. *International Journal of Computer Vision*, 43–77.

Bergen, J. R., P. J. Burt, R. Hingorani, and S. Peleg (1990, December). Computing two motions from three frames. In *Proc. Third International Conference on Computer Vision*, pp. 27–32.

Black, M. J. and P. Anandan (1990, December). A model for the detection of motion over time. In *Proc. Third International Conference on Computer Vision*, pp. 33–37.

Blake, A. and A. Zisserman (1987). *Visual Reconstruction*. Cambridge, MA: MIT Press.

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

Fleet, D. J. and A. D. Jepson (1990, August). Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 77–104.

Gamble, E. and T. Poggio (1987). Visual integration and detection of discontinuities: The key role of intensity edges. AI Memo 970, MIT.

Geman, S. and D. Geman (1984, November). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-6*, 721–741.

Heeger, D. J. (1988). Optical flow estimation using spatiotemporal filters. *International Journal of Computer Vision*, 279–302.

Heitz, F. and P. Bouthemy (1993, December). Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence 15*, 1217–1232.

Hildreth, E. C. (1983). *The Measurement of Visual Motion*. Cambridge, MA: MIT Press.

Horn, B. K. P. and B. Schunck (1981). Determining optical flow. *Artificial Intelligence 17*, 185–203.

Hutchinson, J., C. Koch, J. Luo, and C. Mead (1988, March). Computing motion using analog and binary resistive networks. *Computer 21*, 52–63.

Kanade, T. and M. Okutomi (1994, September). A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Trans. on Pattern Analysis and Machine Intelligence 16*(9), 920–932.

Kaplan, G. A. (1969). Kinetic disruption of optical texture: The perception of depth at an edge. *Perception & Psychophysics 6*(4), 193–198.

Kearney, J. K., W. B. Thompson, and D. L. Boley (1987, March). Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-9*, 229–244.

Koch, C., H. T. Wang, B. Mathur, A. Hsu, and H. Suarez (1989, March). Computing optical flow in resistive networks and in the primate visual system. In *Proc. Workshop on Visual Motion*, pp. 62–69.

Konrad, J. and E. Dubois (1992, September). Bayesian estimation of motion vector fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence 14*, 910–927.

Marr, D. and T. Poggio (1979, October). Cooperative computation of stereo disparity. *Science 194*, 283–287.

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

Mutch, K. M. and W. B. Thompson (1983). Hierarchical estimation of spatial properties from motion. In A. Rosenfeld (Ed.), *Multiresolution Image Processing and Analysis*, pp. 343–354. Springer-Verlag.

Mutch, K. M. and W. B. Thompson (1985, March). Analysis of accretion and deletion at boundaries in dynamic scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-7*, 133–138.

Prazdny, K. (1985). Detection of binocular disparities. *Biological Cybernetics 52*, 93–99.

Rangarajan, K., M. Shah, and D. V. Brackle (1988, December). Optimal corner detection. In *Proc. Second International Conference on Computer Vision*, pp. 90–94.

Schunck, B. G. (1989, October). Image flow segmentation and estimation by constraint line clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-11*, 1010–1027.

Scott, G. L. (1988). *Local and Global Interpretation of Moving Images*. Los Altos: Morgan Kauffman.

Smitley, D. L. and R. Bajcsy (1984). Stereo processing of aerial, urban images. In *Proc. Seventh Int. Conference on Pattern Recognition*, pp. 433–435.

Thompson, W. B. and S. T. Barnard (1981, August). Lower-level estimation and interpretation of visual motion. *Computer 14*, 20–28.

Thompson, W. B., D. Kersten, and W. R. Knecht (1992). Structure-from-motion based on information at surface boundaries. *Biological Cybernetics*, 327–333.

Thompson, W. B., K. M. Mutch, and V. A. Berzins (1985, July). Dynamic occlusion analysis in optical flow fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-7*, 374–383.

Thompson, W. B. and J. S. Painter (1992, July). Qualitative constraints for structure-from-motion. *Computer Vision, Graphics and Image Processing: Image Understanding*, 69–77.

Waxman, A. M., J. Wu, and F. Bergholm (1988). Convected activation profiles and the measurement of visual motion. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, pp. 717–723.

Yonas, A., L. G. Craton, W. B. Thompson, and K. F. Condry (1990, April). Boundary identification and the computation of relative motion: Two processes in kinetic occlusion. *Abstracts in Investigative Ophthalmology and Visual Science 31*(4), 524.