Image Flow Segmentation and Estimation by Constraint Line Clustering

BRIAN G. SCHUNCK, MEMBER, IEEE

Abstract—**Image flow is the velocity field in the image plane caused by the motion of the observer, objects in the scene, or apparent motion. The image flow velocity field is an important intrinsic image and many algorithms that use the image flow velocity field have already been described. The image flow velocity field can contain discontinuities due to object occlusion in the scene. An algorithm that can estimate the image flow velocity field when there are discontinuities due to occlusion is described. Experimental results on a demanding synthetic test case and a real image are presented. Some error analysis that explains the performance of the algorithm is provided. The velocity field estimate can be improved using surface reconstruction between velocity field boundaries. A new surface approximation algorithm that does not blur sharp boundaries is presented.**

Index Terms—**Computer vision, image flow, motion estimation, motion segmentation, optical flow, smoothing, surface approximation, surface interpolation, surface reconstruction**

I. Introduction

THIS paper describes a new algorithm called constraint line clustering for estimating the image flow velocity field. Particular emphasis is placed on motion estimation and segmentation in situations such as random dot patterns where motion is the only cue to segmentation. Experimental results and error analysis are presented. A new smoothing algorithm for improving the velocity field estimate is described. The smoothing algorithm constructs a smooth estimate of the velocity field by approximating a surface between step discontinuities.

A. Image Flow

Image flow is the velocity field in the image plane that arises due to the projection of moving patterns in the scene onto the image plane. The motion of patterns in the image plane may be due to the motion of the observer, the motion of objects in the scene, or both. The motion may also be apparent motion where a change in the image between frames gives the illusion of motion. Image flow is a short-range visual process that is more local than matching methods which work with image features, but not as local as difference picture methods which work with pixel-sized regions [1].

Image flow is modeled by the image flow constraint equation,

$$E_x u + E_y v + E_t = 0, \qquad (1)$$

which models the interaction between the velocity field (u, v) and the local changes in space and time of the image irradiance $E(x, y, t)$. The image flow constraint equation is discussed by Schunck [2], [1].

B. Applications

Potential applications for image flow include structure from motion, egomotion, object tracking, and image compression. Schunck [3] provides a summary of image flow applications. It is possible that the image flow velocity field cannot be estimated with sufficient accuracy for structure from motion or egomotion. Schunck [4] argues that the image flow constraint equation (1) may not be the appropriate model for using motion to estimate scene structure and observer motion. Regardless of how these concerns are answered, the image flow velocity field can be used for segmentation since the velocity field is useful for qualitative purposes [5].

C. Scope of Presentation

The work described in this paper is concerned with developing an algorithm for estimating the image flow velocity field from image input. Since image flow is a short-range process, long-range motions that would require correspondence over large distances are not considered. It is assumed that image flow is an early vision process that precedes (or works in parallel with) feature extracting processes; hence, image flow algorithms based on feature matching are not considered. The work presented in this paper differs from prior work in the attention given to image flow estimation when there are motion boundaries. Particular emphasis is placed on the performance of the image flow estimation algorithm in cases where motion is the only cue for segmentation and the image sequence contains motion boundaries. An example of this situation is the test pattern used by Braddick [6], explained in Section II-E.

D. Summary

Section II summarizes prior work in image flow estimation. Section III discusses image flow models. The polar form of the image flow constraint equation [2], [1] is

Manuscript received August 8, 1986; revised March 31, 1989. Recommended for acceptance by W. B. Thompson. This work was supported in part by the Advanced Research Projects Agency, Office of Naval Research under Contract N00014-80-C-0505.

The author is with the Artificial Intelligence Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109.

IEEE Log Number 8929346.

presented and sampling effects are discussed. Section IV describes the constraint line clustering algorithm for estimating the image flow velocity field and presents experimental results. Section V provides some error analysis for constraint line clustering. Section VI presents a new smoothing algorithm, called surface-based smoothing, for intrinsic images with discontinuities. Surface-based smoothing must be used to improve the velocity field estimate produced by constraint line clustering. Experimental results with surface-based smoothing applied to the results of constraint line clustering from Section IV are presented. The paper finishes with a discussion of conclusions in Section VII.

II. Background

This section reviews prior work in developing algorithms for estimating the image flow velocity field. Work in image flow estimation by regularization and correlation, image flow estimation for image compression, and image flow estimation along contours will be reviewed. Related work in psychology that suggests a performance goal for image flow estimation algorithms will be discussed. Image flow research related to motion boundaries is summarized. A good overview of early work in image flow is provided by Nagel [7].

A. *Image Flow Estimation by Regularization*

Regularization has been used to develop many vision algorithms for early vision processing [8]. Horn and Schunck [9] formulated an optimization problem for estimating image flow. The optimization measure consisted of two terms: a penalty on the deviation of the estimated velocity field from the image flow constraint equation and a penalty on the deviation of the velocity field components from smooth surfaces. The smoothness penalty was the sum of the squares of the magnitude of the gradient of image flow velocity. Horn and Schunck [9] implemented iterative equations for solving these equations and presented examples of the application of the iterative equations to several synthetic images. Unfortunately, the algorithm cannot work in cases where there are motion boundaries in the velocity field since the smoothness constraint leads to iterative equations that blur abrupt changes in the velocity field.

A test case that demonstrates some of the problems with image flow estimation algorithms based on regularization is a textured box translating across a uniform background. The textured box is any rectangular region with a random pattern of image irradiance and the uniform background has constant image irradiance. Suppose that an image flow estimation algorithm based on regularization, such as one developed by Horn and Schunck [9], is applied to this test case for many frames until the object is beyond the field of view. Let a small number (possibly as few as one) iterations of the algorithm be performed per frame of the test sequence. This provides the best results in most cases since new motion constraint information is injected into the algorithm [9, Sect. 17]. In the first few frames of the sequence, an image flow velocity field will be estimated within the boundaries of the box. The velocity field outside of the box will be zero. As the box moves, the region of nonzero velocity field does not move along with the box since there is nothing in the algorithm that forces the velocity field to be extrapolated in the direction of motion. In fact, this would be the wrong thing to do in many cases such as rotation or dilation without translation. Since the location of the region of nonzero velocity field is not forced to move along with the box, the translating box moves out from under the region of nonzero velocity vectors. New nonzero velocity vectors are formed on the surface of the box as it moves into positions. The algorithm does not eliminate the old region of velocity that trails behind the box since the uniform background provides no constraint on the possible velocities. This creates a "comet tail" of velocity behind the box. In regions of the image where the gradient is zero, the Horn and Schunck algorithm degenerates into the Laplace equation which smooths the velocity field. The region of nonzero velocity vectors that trails behind the box is smoothed into a consistent (though obviously wrong) velocity field which persists long after the translating box moves beyond the boundaries of the image. The erroneous velocity field will never be eliminated; it is just gradually smoothed out into the uniform background.

B. *Motion Estimation by Correlation*

Poggio and Reichardt [10], Reichardt and Poggio [11], and Buckner [12], have studied the visual system of the fly and concluded that the behavior of the motion detection system can be described as a correlation. Any correlation scheme for motion estimation, whether implemented in a biological organism or in a machine vision system, cannot produce a velocity field estimate with sharp boundaries. Correlation cannot achieve the type of result desired in this work.

The work of Poggio and Reichardt [10], [11] was based on the Volterra series for modeling systems with multiple inputs [13]. The Volterra series multiplies inputs but provides no special facilities for representing discontinuities.

C. *Motion Estimation for Image Compression*

Haskell [14], Limb and Murphy [15], [16], Cafforio and Rocca [17], Netravali and Robbins [18], Stuller and Netravali [19], Stuller, Netravali, and Robbins [20], and Jones and Rashid [21] have studied motion estimation for image compression. Additional work was reported by Paquin and Dubois [22]. Image compression has not forced the development of image flow estimation algorithms that handle discontinuities because image compression does not require perfect estimation of the motion and does not require the detection of motion boundaries. Any discrepancy between frames caused by inaccurate estimation of the motion is transmitted as a correction. It may be true that a better motion estimation algorithm could reduce the image bandwidth further, but any improvement must be balanced against the added cost of the image compression implementation [3].

D. Image Flow Estimation Along Contours

Recent work used edge information in the flow computation by combining motion information along edge contours [23]-[26]. There are problems with this approach: 1) the velocity vector of an edge is a very powerful clue for grouping edges into contours, but this advantage is dismissed when motion estimation proceeds contour grouping, 2) edge fragments can be combined into contours that cross motion boundaries and this will lead to an incorrect motion estimate, and 3) it is not necessary to restrict motion information to edges since it is possible to develop one algorithm that simultaneously estimates the motion at edges and interpolates between edges. The motion estimation algorithm developed by Horn and Schunck [9] uses a single set of iterative equations that, in effect, automatically balance the computations performed in regions of high and low gradient. When the gradient is large, the information provided by the image flow constraint equation dominates the smoothing process; when the gradient is small, less weight is given to the motion constraint and the iterative equations reduce to the computation of a surface approximation determined by the form of the smoothness constraint. The surface approximation computation iteratively fills in the velocity field using the estimates derived from areas where the gradient is large. Another problem with image flow estimation algorithms that follow edge contours is that they do no better in situations where there are velocity field discontinuities than any of the other algorithms summarized in this section.

E. Results from Psychology

Although it was not the intent of this work to develop a model for human motion perception, some results from psychology focused this project on solving a demanding problem in motion segmentation with translating objects [6], [27]-[29]. The work of Braddick [6] provided the key test case and standard of performance for this research. Braddick [6] performed apparent motion experiments with pairs of random-dot images. Pixels were randomly chosen to be black or white. A central rectangle within one image was displaced to form the second image. No texture of image intensity boundaries were present in the test images. The image pairs used by Braddick differed in pixel size and displacement distance. Subjects were required to decide whether the rectangle was horizontal or vertical and rate the clarity of the motion percept. Performance and clarity decreased with the size of displacement, irregardless of the pixel size. For small displacements of around 5′ of arc, subjects reported that the motion was coherent and the boundaries of the rectangle could be clearly seen. The range over which the clarity of the motion perception deteriorates corresponds closely with the increasing sizes of the channels in the edge detection theory of Marr and Hildredth [30]. As the displacement increases, it is within the support of fewer channels and the ability to perceive coherent motion decreases. Beyond 20′ of arc, coherent motion was not perceived and the bounding contour of the rectangle was not clearly seen. Braddick argued that if human vision used an algorithm that matched pixels of one type (white or black) to nearby pixels of the same type, then when the size of the displacement was more than one pixel subjects would not see coherent motion. Since this did not occur, Braddick argued that human vision must incorporate a more sophisticated algorithm than matching nearby pixels to perceive coherent motion. Subsequent analysis [29, p. 18] argues for a short-range process based on changes in image intensity in addition to a long-range process based on feature correspondence.

This research is not concerned with discovering the mechanisms of biological vision systems, but the results from psychology on the short-range motion process were essential in focusing this project. Since the short-range motion process in human subjects can detect motion boundaries in scenes where other cues are not present, it must be possible to develop an algorithm for estimating the image flow velocity field while retaining the sharp changes in velocity values that may occur across a motion boundary that corresponds to occlusion of scene surfaces. The range of displacement over which coherent motion is perceived in the short-range motion process is limited by the size of the largest channel. In machine vision, the maximum displacement for which an image flow estimation algorithm can be applied is limited by the size of the smoothing filter applied to the image or the size of the aperture function if no smoothing is performed. The aperture function is usually approximately equal to the pixel spacing. If no smoothing filter is applied to the image, then the displacement between frames must be no greater than the pixel spacing.

In summary, the illusion of Braddick [6] is a key test case for image flow estimation because it requires estimation and segmentation based on short-range motion without other visual cues. Since the subjects in the experiments conducted by Braddick were able to clearly perceive the outline of the moving rectangle, an image flow estimation algorithm should be able to retain sharp step changes in the velocity field when given a random dot image pair as input.

F. Image Flow Estimation and Motion Boundaries

Nakayama and Loomis [31] proposed a function called the convexity function for detecting motion boundaries in the image flow field, but did not explain how the velocity field could be estimated in the vicinity of the motion boundary. Batali and Ullman [32] tried a local consistency check on the motion information provided by directional selectivity to segment random binary images, but the algorithm did not estimate the velocity field. Mutch and Thompson [33] presented a correspondence algorithm for motion analysis that detected occlusion by looking for significant regions where matches could not be found. Thompson, Mutch, and Berzins [34] developed a motion detection operator based on the zero-crossing detector of Marr and Hildreth [30] and developed an analysis of motion boundaries that allowed the occluding surface to be differentiated from the occluded surface.

Fennema and Thompson [35] developed an algorithm for estimating the velocity vector of the most prominent object in the scene by using histogram analysis to combine image flow constraints. The algorithm could only determine the motion of the single most prominent object, could not handle rotation or translation in depth, and assumed that the background was stationary. Since the local structure was lost, the algorithm could not compute an image flow velocity field and this meant that the algorithm was not useful for scenes containing multiple objects. For example, it is not possible to differentiate between 1) the situation where one large object is translating and 2) the situation where two objects, each half the size of the object in situation (1), but separated by some significant distance, are translating. Nevertheless, the algorithm was significant because the histogram technique was immune to the incorrect motion constraints generated along motion boundaries [1] and was not confused by multiple moving objects in the scene as long as one moving object was significantly larger than the others. The Fennema and Thompson algorithm inspired the cluster analysis approach presented in this paper.

G. Summary of Prior Work

Image flow algorithms have been developed with regularization, but the algorithms are not suitable for real-world scenes because the algorithms cannot handle velocity field discontinuities. Correlation and image flow estimation along contours have the same problem. Much work on image flow estimation has been done for image compression, but the problem of estimating the image flow velocity field with discontinuities has not been addressed. Results from psychology strongly suggest that the human vision system can estimate the image flow velocity field without blurring motion boundaries. This implies that a machine vision system should be capable of estimating the image flow velocity field with little distortion in the velocity field discontinuities. The intent of this work is to develop an image flow velocity field estimation algorithm with performance comparable to that of the human vision system; the algorithm should be able to estimate the velocity field in images that contain motion boundaries without blurring the boundaries or allowing the presence of the boundaries to distort the velocity field estimate. It is not the intent of this work to propose the image flow estimation algorithm presented in this paper as a model for the human vision system, since it would require more experimentation and comparison with psychophysical results; but the algorithm presented in this paper may suggest a model for image flow estimation in the human vision system that could be subjected to the required tests.

III. Image Flow Models

The image flow constraint equation (1) models the relationship between the velocity field and the changes in the image irradiance in space and time. The polar form of the equation is the basis for the algorithm development in this paper. As stated by Schunck [1], [4], two conditions are required to ensure the validity of the image flow constraint equation: 1) the perceived change in image irradiance at each point in the image plane must be entirely due to motion of the image pattern, as opposed to changes in the pattern due to reflectance effects, and 2) the image must be smooth except at a finite number of discontinuities. The first condition is satisfied by the image of a translating, diffuse object with distant light sources; the second condition is a technical point. In other words, the perceived change in image irradiance at each point in the image plane must be entirely due to shifts of the image pattern, as opposed to changes in the pattern. These conditions are probably too restrictive to allow image flow to be useful for structure from motion, but the image flow equation is obeyed in practice with sufficient accuracy for tasks such as segmentation.

A. Polar Form of the Constraint Equation

The image flow constraint equation defines a line in velocity space as shown in Fig. 1. The line in velocity space that is the locus of possible velocities specified by the image flow constraint equation is called the constraint line. The constraint line is uniquely defined by d, the distance of the line from the origin along the perpendicular bisector and the angle α with respect to the u axis. The displacement and angle of the constraint line are given by

$$d = \frac{|E_t|}{\sqrt{E_x^2 + E_y^2}} = \frac{|E_t|}{|\nabla E|}; \tag{2}$$

$$\alpha = \begin{cases} \arctan (E_x, E_y) & \text{if } E_t \geq 0; \\ \arctan (-E_x, -E_y) & \text{otherwise.} \end{cases} \tag{3}$$

To derive the constraint equation in polar coordinates, note that the image flow equation contains the dot product of the image irradiance gradient with the velocity vector

$$E_x u + E_y v + E_t = \nabla E \cdot (u, v) + E_t. \tag{4}$$

If ρ is the speed of motion, α is the angle of the constraint line, and β is the direction of motion, then the dot product in (4) is

$$\rho |\nabla E| \cos (\alpha - \beta). \tag{5}$$

Dividing through by the magnitude of the image gradient yields the polar form of the image flow constraint equation

$$d = \rho \cos (\alpha - \beta). \tag{6}$$

Note that β is constrained to be between $\alpha - \pi/2$ and $\alpha + \pi/2$. Since the displacement d of the constraint line from the origin must always be nonnegative, the orientation α is reflected when $E_t > 0$. The displacement d is the projection of the motion vector onto the line of the gradient of image irradiance and is independent of the magnitude or polarity of the gradient.

The polar form of the image flow constraint equation

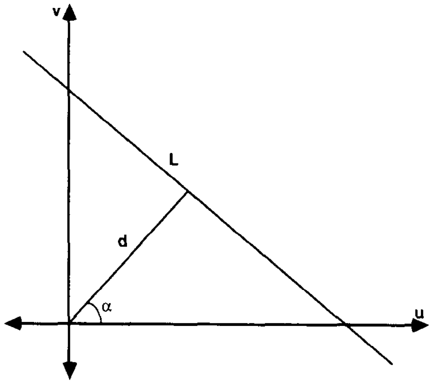

Fig. 1. The locus of points satisfied by the image flow constraint equation is shown. The equation defines a line of points in velocity space with velocities that satisfy the constraint. The distance d from the origin to the constraint line L is called the displacement of the constraint line; the angle α is called the orientation of the constraint line.

should be used in the analysis of algorithms for image flow estimation with motion boundaries since the polar form will not contain δ-functions at step discontinuities. In other words, if an example contains an ideal step discontinuity in image irradiance, then the coefficients in the conventional form of the image flow constraint equation (1) will be δ-functions. The polar form will not contain δ-functions since the displacement d is the ratio of the magnitude of the change in time of the image irradiance to the magnitude of the gradient of the image irradiance and this ratio remains finite in the limit as the image irradiance becomes an ideal step discontinuity.

B. Sampled Image Flow Models

The time-varying image $E(x, y, t)$ is sampled in space and time to produce an image sequence. The spatial sampling distance must be smaller than the scale of image texture. This concept is most easily explained by refering to Fig. 9. Note how the shift in the image irradiance function is small relative to the wavelength of the image irradiance. The temporal sampling period must be much shorter than the scale of time over which the velocity field changes and sufficiently short, relative to the velocity, so that the local displacement is smaller than the local scale of the image texture. In other words, the image irradiance pattern in the image sequence must evidence a phase shift that is large enough to be measured accurately and small enough to avoid phase ambiguity.

The constraints are sufficient to allow the sampled image sequence to accurately portray the local motion data between motion boundaries, but are not sufficient to capture the motion information at motion boundaries. The motion constraints computed along motion boundaries are extremely sensitive to the effects of occlusion boundaries and are usually severely in error. No practical sampling period can provide useful motion information along a motion boundary [2], [1]. An image flow estimation algorithm must be able to work in situations where there are motion boundaries in the image; consequently, the algorithm must be resistant to grossly incorrect measurements. The errors along motion boundaries are too severe to ignore.

In this work, any motion constraint that deviates greatly from the true motion constraint is called a grossly incorrect constraint. Experimental observations [2], [1] indicate that grossly incorrect constraints almost always occur along motion boundaries. Since motion estimation in the presence of motion boundaries is central to this work, grossly incorrect constraints must be handled.

IV. Constraint Line Clustering

An algorithm called constraint line clustering [36] that estimates the image flow field when the image irradiance pattern or the velocity field contains discontinuities will be presented.

A. Problem Statement

The constraint line clustering algorithm uses the polar form of the image flow constraint equation

$$d = \rho \cos(\alpha - \beta) \tag{7}$$

where $\rho(x, y)$ and $\beta(x, y)$ are the speed and direction of motion, respectively. The velocity vector (ρ, β) for the motion at any point in the image must lie along the line in velocity space defined by the image flow constraint equation (7) shown in Fig. 1. The constraint line is uniquely defined by the displacement d of the constraint line from the origin and the orientation α of the constraint line. The constraint line displacement has the dimensions of speed; it is the minimum speed consistent with the motion constraint.

Assume that the image intensity $E(x, y, t)$ already incorporates any spatial or temporal filtering performed on the image. For example, the image could be smoothed with a Gaussian filter. The image is sampled in space and time to produce an image sequence $E(x_i, y_j, t_k)$. The d and α intrinsic image arrays $d(x_i, y_j)$ and $\alpha(x_i, y_j)$ are computed from the image sequence using the formulas in (3). The partial derivatives are computed by first differences over a cube in space and time [9, sect. 7]. The image flow estimation problem is to compute the intrinsic image arrays $\rho(x_i, y_j)$ and $\beta(x_i, y_j)$ for the speed and direction of motion from the motion measurements in the d and α arrays. Prior knowledge of the image flow velocity field is not available. By assumption, the image flow equation (7) describes the changes in the image sequence; hence, the restrictions discussed by Schunck [1] apply.

B. Constraint Line Clustering

The constraint line clustering algorithm uses a clever form of cluster analysis to extract the motion estimate from contradictory data. What could be a multidimensional cluster analysis problem is transformed into a trivial cluster analysis problem in one dimension. Suppose that for each d and α measurement a set of measurements $\{d_i, \alpha_i\}$ is taken from some spatial neighborhood of the point that generated d and α. Compute the set of intersections of each of the neighboring measurements d_i and α_i with the given d and α measurement. All of these intersections lie along the line defined by d and α. Any constraint lines in $\{d_i, \alpha_i\}$ that are part of the same region of motion as d and α will tend to intersect the line defined by d and α in a tight cluster around the true velocity. Any constraint lines in $\{d_i, \alpha_i\}$ that are from regions of different motion will intersect the line defined by d and α over a broad range of positions. The grossly incorrect constraint lines generated along a motion boundary will not intersect the line defined by d and α at a consistent point. The velocity estimate for the neighborhood of a given d and α measurement can be computed by one-dimensional cluster analysis of intersections along a constraint line.

1) Computing Constraint Line Intersections: The formula for the position of the intersection along a constraint line is easily derived by first rotating the coordinate system so that the constraint line defined by d and α is parallel to the vertical axis in velocity space. Let the angle between the two constraint lines be denoted by $\phi = \alpha' - \alpha$, as shown in Fig. 2. The distance of the intersection along the constraint line defined by d and α is given by

$$b = \frac{c}{\tan \phi}. \tag{8}$$

The distance c is related to d, d', and ϕ by

$$(d + c) \cos \phi = d', \tag{9}$$

so using (8) to eliminate c from (9) yields

$$d \cos \phi + b \sin \phi = d'. \tag{10}$$

Solving for b and substituting the definition for ϕ yields

$$b = \frac{d' - d \cos \phi}{\sin \phi} = \frac{d' - d \cos (\alpha' - \alpha)}{\sin (\alpha' - \alpha)}. \tag{11}$$

This formula provides the position along a constraint line defined by d and α of the intersection of the constraint line with another constraint line defined by d' and α'.

2) One-Dimensional Cluster Analysis: Given a set $\{b_i\}$ of intersections of the constraint lines within a neighborhood with a given constraint line at the center of the neighborhood, the set of intersections must be analyzed to determine the most consistent subset of intersections that cluster about the likely velocity. The cluster analysis criterion is motivated and explained by the following example. Imagine that a motion boundary passes almost vertically through a neighborhood just to the left of the center element. At any reasonable pixel resolution, the boundary will most likely pass smoothly through the neighborhood dividing the neighborhood almost in half. The region on the left corresponds to one surface in the scene and the region on the right including the center element corresponds to a different surface. In the extreme case where the boundary passes very close to the center of the neighborhood, almost half of the intersections of neighborhood constraint lines with the constraint line from the center of the neighborhood will be with constraints from the left surface or the motion boundary itself. These intersections will most likely not be close to the velocity of the right surface from which the center constraint line was obtained. Almost half of the intersections may be useless in the extreme case; but at least half of the intersections will correspond to the same region of motion. Since rejecting correct intersections is less harmful than accepting incorrect intersections, a conservative stand is taken: the algorithm looks for the tightest cluster that contains roughly half of the intersections.

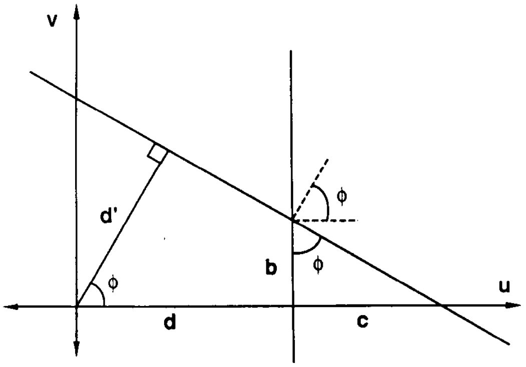

Fig. 2. The method for deriving the formula for the position of the intersection of two constraint lines along one of the constraint lines is shown.

In one dimension, cluster analysis is easy since there is a total ordering of points along a line and cluster analysis reduces to interval analysis. The constraint line clustering algorithm sorts the set of n intersections $\{b_i\}$ and then looks for the tightest interval that contains half of the intersections by examining successive pairs of intersections that are $\lfloor n/2 \rfloor$ intersections apart. The algorithm chooses the pair of intersections that are closest together as the estimate of the majority cluster of intersections. The test assumes that there is at most one motion boundary within the spatial neighborhood of the d and α measurement and the motion detectors are sufficiently finely spaced that the motion boundary within the neighborhood does not pass erratically through the neighborhood. The assumption excludes cases where the center d and α measurement in a neighborhood is from one side of a motion boundary while the majority of d and α measurements in the neighborhood are from the opposite side of the motion boundary. In practice, the constraint line clustering algorithm continues to work when the assumption is violated for reasons explained in Section IV-C1).

A formula for the exact fraction of constraint lines that should be in the majority when the motion boundary passes cleanly through the neighborhood is easily derived. Suppose that a straight, vertical motion boundary passes through a neighborhood to one side of the center element. Let the neighborhood size be n by n where $n = 2k + 1$ for some positive integer k. The number of constraint lines in the compatible majority is $(2k + 1)(k +$

1), the number of incompatible minority constraint lines is $(2k + 1)(k - 1)$, and the number of erroneous constraint lines is $2k + 1$. The number of compatible constraint line intersections forming a relatively tight cluster will be

$$(2k + 1)(k + 1) - 1 = 2k^2 + 3k = k(2k + 3), \tag{12}$$

and the fraction of constraint line intersections that should form a tight cluster is

$$\frac{k(2k + 3)}{(2k + 1)^2 - 1} = \frac{k(2k + 3)}{4k^2 + 4k} = \frac{2k + 3}{4k + 4}. \tag{13}$$

The constraint line clustering algorithm determines the tightest cluster of constraint line intersections that contains this fraction of the total number of constraint line intersections. There may be fewer than $4k(k + 1)$ intersections, since some of the constraint lines may be parallel to the constraint line at the center of the neighborhood. The location of the center of the shortest interval that contains the fraction of intersections given by the formula above is used as an estimate for the most consistent velocity within the neighborhood. Note that the formula for the fraction of the constraint lines on the majority side of the boundary asymptotically goes to 1/2 as $k \to \infty$.

3) Estimating the Velocity Vector: The center position $\hat{b}$ of the tightest cluster along the constraint line is the midpoint of the smallest interval that contains roughly half of the intersections. The estimated speed of motion is given by

$$\hat{v} = \sqrt{d^2 + \hat{b}^2} \tag{14}$$

and the estimated direction of motion is given by

$$\hat{\beta} = \alpha + \tan(\hat{b}/d). \tag{15}$$

The estimated speed and direction of motion $(\hat{v}, \hat{\beta})$ computed with constraint line clustering from the motion parameters $\{d_i\}$ and $\{\alpha_i\}$ satisfy the problem statement in Section IV-A.

C. Potential Problems with Constraint Line Clustering

Some concerns that may be raised about the performance of constraint line clustering will be addressed in this section. Implementation details of the constraint line clustering algorithm will also be explained.

1) Motion Boundary Corners: The constraint line clustering algorithm is very robust and works even if the basic assumption that at least half of the intersections must be from the same region of motion is violated. For example, if the center constraint is from just inside a right angle corner so that only about 1/4 of the intersections are from the same region of motion, the algorithm still works because the intersections from constraints outside the corner are almost uniformly distributed along the constraint line and do not bias the estimate of the tightest cluster [3]. The error analysis in Section V attempts to explain this in detail.

2) The Aperture Problem: Marr and Ullman [37] discuss the aperture problem for image flow estimation. The problem occurs in any situation where an estimate must be obtained from data in a local neighborhood in an image that may contain a boundary. If an algorithm contains an assumption that the data are consistent over the neighborhood and the assumption is violated, then the image flow estimate will be incorrect. On the other hand, it is necessary to use measurements in a neighborhood large enough to provide sufficient variation in gradient orientation. The constraint line clustering algorithm does not suffer from an aperture problem, because the one-dimensional cluster analysis computation provides a statistical check that prevents the aperture problem from occurring within a neighborhood of any size. This means that the size of the neighborhood can be increased to include more good constraint lines and improve the local estimate of the velocity field. This property is particularly useful given the problem that erroneous constraint lines will occur along a motion boundary. Since the aperture problem is not a concern, the relative number of erroneous constraint lines in a neighborhood can always be controlled by varying the size of the neighborhood. Constraint lines that occur along the curve of a motion boundary are grossly incorrect due to sampling effects [2], [1]. If the neighborhood size w is defined to be the width of a neighborhood, then the number of grossly incorrect constraint lines generated along a motion boundary within a neighborhood increases at most linearly with neighborhood size. The number of constraint lines that are not grossly incorrect always increases as the square of neighborhood size. The ratio of grossly incorrect to reasonably accurate constraint lines will always decrease at least as fast as $1/w$ where w is the neighborhood width. Within the neighborhood of any good constraint line, there will always be enough reasonably accurate constraint lines to form a tight cluster.

3) Numerical Condition: Constraint line intersections are not always numerically well conditioned. When two constraint lines are close in orientation there can be a large error in the intersection position. When the constraint lines have the same orientation and different displacements, the lines will not intersect. In spite of these objections, the constraint line clustering algorithm is robust by virtue of the fact that only the tightest group of points are retained to compute the motion estimate. When an intersection position is computed for two constraint lines that are close in orientation and displacement, then the position of the intersection may be distant from the true position. Nearly half of the interaction positions will be discarded. Any grossly incorrect intersections will most probably be discarded since they lie outside the tightest cluster.

4) Selecting the Neighborhood Size: The algorithm is relatively insensitive to the choice of neighborhood size. Good results have been obtained even for the minimum neighborhood size of 3 by 3. Larger neighborhood sizes produce only slightly better motion estimates. There are two reasons for this. The first reason is that the accuracy of the motion estimate is limited by the accuracy of the constraint line along which the motion estimate is obtained. The second reason is that the number of points in the neighborhood may have little impact on the quality of the motion estimate; it is the spatial extent of the neighborhood relative to the gradient variation that determines the quality of the motion estimate.

Increasing the spatial extent of the neighborhood, without necessarily increasing the number of points in the neighborhood, may bring new information (new constraints) into the calculation of the velocity estimate if the larger neighborhood provides more gradient variation. Consider a region in the image where the gradient is nearly constant. The motion constraints derived in this region will be similar and will not yield good intersection measurements because the intersection computations are ill-conditioned. In order to get good intersection measurements, motion constraints must be used from beyond the region of similar gradient. The motion estimate can be greatly improved by increasing the spatial coverage of the neighborhood. It is not necessary to increase the number of motion constraints in the neighborhood since the motion constraints close to the center of the neighborhood are redundant. The amount of spatial variation in the motion constraint measurements determines the minimum spatial extent of the neighborhood.

The constraint line clustering algorithm has a wide tolerance to the neighborhood size: the neighborhood must be large enough to provide sufficient variation in image gradient direction as explained in Section IV-C6), but small enough so that constraint line clustering does not average away variations in the velocity field as explained in Section IV-C5). The wide tolerance in the neighborhood size made it easy to select the neighborhood size for all of the experiments in this work: a neighborhood size of 5 by 5 possibly with some downsampling as explained in Sections IV-C7) and IV-D) was sufficient.

5) Variation in the Velocity Field: The amount of slow spatial variation in the velocity field may limit the maximum neighborhood size due to an averaging effect. Suppose that a neighborhood does not contain a motion boundary. The intersection measurements will be closely grouped together, relative to the situation where the neighborhood extends across a motion boundary. If there is some variation in the velocity field within the neighborhood, then the spread of the intersection positions will be greater than if there were no variation. There is actually no point where constraint line clustering will fail to yield a reasonable estimate of the center position of the group of intersections along the constraint line from the center of the neighborhood. Using the center position of the intersections as the velocity estimate is like averaging or low-pass filtering the velocities in the neighborhood. The averaging will reduce the spatial resolution of the velocity field estimate. The neighborhood size should be chosen so that the spatial extent of the neighborhood is at least as large as the scale of the variation in the image gradient, but not so large that the resolution in the velocity field estimate would be excessively reduced due to averaging.

6) Insufficient Variation in Image Gradient: Any algorithm for estimating the image flow velocity field depends on the assumption that there will be sufficient constraints on the motion. If there are insufficient constraints, then the algorithm cannot solve the problem; and if the constraints conspire to yield an incorrect solution, then the algorithm will not compute the correct solution. For example, if all of the constraint lines in the neighborhood of a given constraint line are nearly parallel except for the constraint line at the center of the neighborhood, then the bundle of constraint lines may not intersect the center constraint line near the correct velocity. The point is that any algorithm including regularization over regions [9] or regularization along contours [24], [25] will have trouble in cases where there is insufficient or misleading motion constraints because fundamentally any algorithm will have trouble if the data is ill-conditioned. In other words, any algorithm including constraint line clustering must have sufficient variety of information. Since constraint line orientation is gradient direction or its opposite, there must be sufficient variety of gradient directions in the neighborhood.

In practical examples, there were spurious problems in stationary areas of the test images. Some points in the velocity estimate that should have had velocities near zero had large velocities. The constraint line clustering algorithm was not at fault. The constraint line displacements at the points corresponding to these erroneous velocity estimates were large when they should have been very small. Closer inspection revealed that the magnitude of the image gradient was very small at the points where the large d values occurred. The errors in the constraint line displacements were caused by numerical ill-conditioning. The constraint line displacement is computed by

$$d = \frac{|E_t|}{|\nabla E|} \tag{16}$$

and the calculation is most sensitive when $|\nabla E|$ is small. To eliminate this problem, the constraint line displacement is forced to zero whenever the magnitude of the image gradient is small. A small gradient indicates locally uniform image irradiance and the velocity estimate in a region of uniform irradiance should be zero. Changing the constraint line displacement from an erroneously large value to zero allows constraint line clustering to produce a small velocity if the majority of the constraint lines in the neighborhood also have small displacements.

7) Downsampling: If the width of the neighborhood over which the constraint line clustering is performed is not large enough to provide a set of constraint lines of sufficient variety to allow the constraint line clustering algorithm to form a well-conditioned set of intersections, then the size of the neighborhood over which constraint line clustering is performed can be increased. Because of the statistical nature of constraint line clustering, there will be no problem if the larger neighborhood extends across a motion boundary. In practice, a large neighborhood may not be desirable since the amount of computation that must be performed is proportional to the area of the neighborhood. Excessive computation may be avoided by downsampling the arrays of d and α values prior to applying the constraint line clustering algorithm. Downsampling by a factor of k increases the effective area of the constraint line clustering neighborhood by a factor of k^2 without increasing the amount of computation that must be performed for each neighborhood. The total amount of computation decreases by a factor of k^2 since the constraint line clustering is performed over fewer neighborhoods in the d and α arrays.

An alternative to downsampling the d and α arrays is to use overlapping sparse neighborhoods. The constraint line clustering neighborhoods could be extended to cover the same pattern of points as in downsampling, but neighborhoods would be overlapped into the points that were left unused in pure downsampling. The amount of computation performed per neighborhood would be the same and the effective area of the neighborhood would increase in the same manner as for downsampling. But the neighborhoods would be overlapped and the number and density of velocity measurements obtained would be about the same as the size and density of the original d and α arrays.

8) Incorrect Center Constraint Line: One apparent weakness with constraint line clustering is that it depends on the accuracy of the constraint line at the center of the neighborhood. In fact, this is not a problem with the constraint line clustering algorithm itself, but is an instance of a more fundamental problem with assigning a motion estimate to a point in an image. If the motion constraint information is grossly incorrect, then there is a fundamental ambiguity in assigning a motion estimate to the corresponding image location. For example, if the grossly incorrect motion constraint is caused by a motion boundary (which is likely [2], [1]), then it is unclear which motion present in the neighborhood should be assigned to that location. If the grossly incorrect motion constraint is not caused by a motion boundary, then it is still caused by some effect in the image and the ambiguity is fundamental and legitimate. Stated briefly, if the motion constraint is grossly incorrect, a correct motion estimate cannot be assigned to the location. Another way of saying this is that d and α can be computed accurately when the gradient is large and there is no motion boundary; otherwise, no motion estimate is reasonable.

9) Relationship to Other Methods: The information in the image flow constraints could be combined by least squared methods [38]–[41], if it were not for the focus of this research on segmentation in situations where there are several motion boundaries. The image flow constraint equation is only one constraint with two unknowns and the problems in combining image flow constraints near motion boundaries are so severe that least-squares algorithms cannot be used without modifications to select a consistent set of motion constraints [2], [1]. Fischler and Bolles [42] present an example where the least-squares fit of a model to inconsistent data leads to a wrong estimate that cannot be fixed by eliminating the outlier. In this problem, the constraints that are far from the true velocity are the outliers. Robust statistics provides some methods for fitting models to inconsistent data [43], [44]. These methods were not used in this work because the special structure of the problem leads to a more efficient solution. Given the assumption on motion boundaries used in constraint line clustering, the test for outliers is simple. Further experiments should be conducted to compare robust estimates with constraint line clustering.

There are several kinds of robust statistical measures. In typical robust statistical procedures, data points are sorted into ascending order and statistical measures are derived from the ordering of the data points. Two measures that may be related to this work are M-statistics and L-statistics. An M-statistic is a maximum likelihood estimate derived from the order of a set of measurements [44, pp. 43–55]. An L-statistic is a linear combination of the ordered data points [44, pp. 55–61]. An example of an L-statistic is the range of a set of data points [43, p. 33]. Constraint line clustering informally combines several ideas from robust statistics. The selection of the tightest interval that contains half of the data points resembles an M-statistic; but there is no formal proof that the tightest interval containing half the data points is the most likely choice. Once the tightest interval has been determined, the location of the velocity estimate is computed with an L-statistic. Constraint line clustering also resembles the trimmed mean for computing the mean while eliminating the influence of outliers [45, p. 247]. Order statistics [43] presumes that the data points can be ranked into ascending order. In constraint line clustering, the data points that are ranked are the locations along the center constraint line of the intersections with neighboring constraint lines. Although constraint line clustering achieves good results, the intersections may not be the best choice as a starting point for the application of formal statistical methods. Further work should be done with robust regression for determining the velocity field estimates directly from the constraint line coefficients.

The random sample consensus (RANSAC) algorithm [42] developed by Fischler and Bolles handles outliers by computing an estimate from a consistent subset of inconsistent data points. The algorithm randomly selects a set S of the minimum number of points required to fit a model from the set of data points P. The set S is used to compute a model estimate. The consensus set S^* of points in P that are within an error tolerance τ of the model is determined. If the number of points in S^* is greater than some threshold, then the model is recomputed using the data points in S^*; otherwise, a new set S is randomly selected from the data points P and the consensus procedure is repeated. After a predetermined number of iterations, the algorithm either fails or settles for the model obtained from the largest consensus set. The RANSAC algorithm has three parameters that must be chosen by the user: the error tolerance, the minimum acceptable size of the consensus set, and the number of trials. In constraint line clustering, the consensus set is the set of intersections in the tightest interval. There is no error tolerance that must be chosen by the user and the size of the consensus set is fixed at half of the intersections. The fixed limit on the size of the consensus set is justified by the worst case geometry of a motion boundary in a neighborhood which in turn is justified by reasonable assumptions on pixel spacing as described in Section IV-B2). The user does not have to choose the limit on the number of trials in constraint line clustering because the algorithm is not iterative. Although constraint line clustering shares many important similarities with random sample consensus, constraint line clustering avoids the choice of parameters and is faster since it it not iterative. Constraint line clustering is like a one step RANSAC algorithm that eliminates parameter choices because it is tailored to the specific problem of estimating the image flow velocity field in situations that satisfy the restrictions of the short-range motion process.

In principle, a random sample consensus algorithm could be developed for this problem by randomly choosing pairs of constraint lines until an intersection is found that is consistent with at least half of the constraint lines; but an iterative algorithm would be more computationally demanding than constraint line clustering and it would be very hard to choose the error tolerance for the consensus set. Experimental observations made during this work indicate that the width of the tightest interval in constraint line clustering can vary greatly. In the early stages of testing the constraint line clustering algorithm, it was thought that the width of the tightest interval could be used as a measure to detect motion boundaries; but it was found that correct velocity field estimates could have intervals that were very wide and it was not possible to choose a threshold on interval width that would identify boundary points.

D. Experimental Results

A version of the test stimulus used by Braddick [6] was the key test case for developing the constraint line clustering algorithm for image flow estimation. A synthetic image was generated with a textured square on a textured background. The texture of the foreground square and the background were produced by independent sampling from a uniform random number generator in the range [0, 256). The foreground image was overlayed on the background image at a position roughly in the center of the background for the first frame of the test case and at the position one pixel to the right for the second frame of the test case. Uniform noise at various amplitudes was added to these test frames to simulate realistic and greater than realistic noise levels. The additive noise was obtained by independent sampling from a uniform random number generator in the range $[-\mathfrak{N}, \mathfrak{N}]$, where the noise amplitude $\mathfrak{N}$ was some specified percentage of the maximum pixel amplitude which was 255. After adding a noise sample to a pixel, the pixel was compared to the original pixel range of [0, 256). If the pixel was negative, then it was changed to zero; otherwise if the pixel was greater than 255, then it was reduced to 255. This procedure models saturation in black and white. Subsequent work with images obtained from an RCA vidicon camera indicated that a noise level at 5 percent of the maximum pixel amplitude simulated typical error magnitudes. Since the foreground and background textures had identical statistics, the square could not be seen in either frame alone. Only motion between frames allowed the square to be differentiated from the background. The two frames of this test case are displayed in Fig. 3.

The results of applying constraint line clustering with 5 by 5, 9 by 9, and 13 by 13 neighborhoods to the displaced square test case with 5 percent uniform noise is displayed in Figs. 4, 5, and 6, respectively. The important point to note is that the estimated velocity field is not improved by increasing the size of the neighborhood in this test case. This means that the quality of the motion estimate is limited by the quality of the d and α measurements, not by the performance of the constraint line clustering algorithm. This is an attractive characteristic from the stand point of this research since it indicates that the algorithm will work well with a given accuracy for the d and α measurements. Better algorithms for measuring the d and α data will lead to better estimates of the image flow velocity field, but constraint line clustering works very well with the primitive methods for computing d and α used in this work.

Constraint line clustering was run on a real image pair shown in Fig. 7. The scene was taken with a camera looking down on a laboratory table. A sheet of circuit board was lying on the table and a phonebook with an outdoor scene was placed on the circuit board. Scene illumination was provided by typical fluorescent room lighting. The camera was an inexpensive RCA Vidicon camera with automatic gain control that precluded any adjustment of the lens aperture. The phonebook was displaced by hand slightly between frames. The displacement was smaller than the wavelength of the texture on the cover of the phonebook as shown by the slices through the original image pair in Fig. 8. The test images were very noisy as can be seen from the plots in Fig. 8. Each image was smoothed with a 16 by 16 point Gaussian filter. Three slices after smoothing are shown in Fig. 9. The displacement between the slices shows the degree of displacement appropriate to a motion estimation algorithm based on the image flow constraint equation. Even after Gaussian filtering, the quality of the motion constraint data was very poor as can be seen by looking at the scatter plot of constraint line displacement versus angle in Fig. 10. Ideally, the data should form a cosine pulse where the speed of motion is the height of the pulse and the direction of motion corresponds to the center of the pulse; but errors due to noise and occlusion are very severe. To reduce computation and increase the spatial extent of the constraint line clustering neighborhood, the d and α arrays were

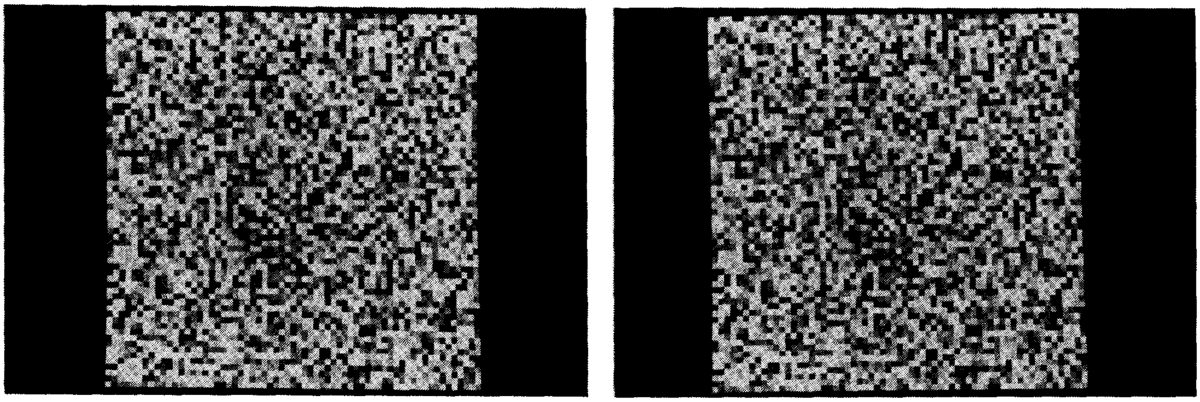

Fig. 3. The two frames of a test case used frequently in this work are displayed. The test frames contain a central square that is displaced one pixel to the right between FRAME-1 and FRAME-2. This frame pair does not contain added noise.

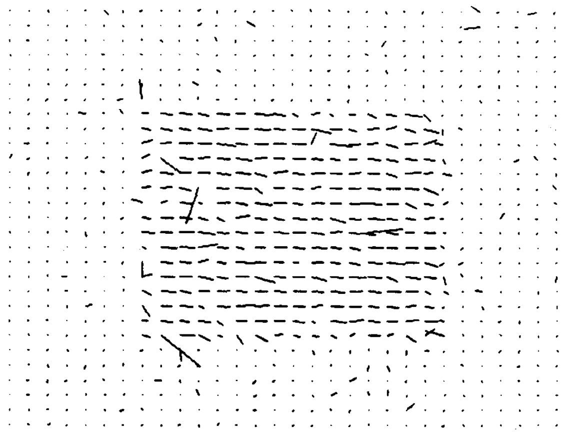

Fig. 4. The velocity field after applying constraint line clustering to the test case of the displaced square with 5 percent uniform noise added is displayed. The neighborhood size was 5 by 5.

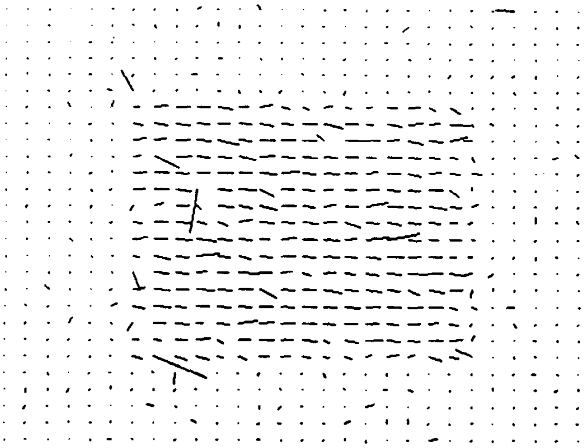

Fig. 5. The velocity field after applying constraint line clustering to the test case of the displaced square with 5 percent uniform noise added is displayed. The neighborhood size was 9 by 9. Note that there is no significant improvement over the field displayed in Fig. 4.

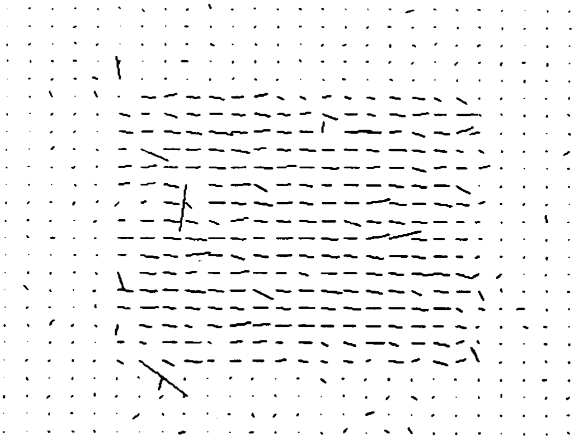

Fig. 6. The velocity field after applying constraint line clustering to the test case of the displaced square with 5 percent uniform noise added is displayed. The neighborhood size was 13 by 13. Note that there is no significant improvement over the field displayed in either Fig. 4 or Fig. 5. The constraint line clustering algorithm performed very well; the quality of the motion estimate was limited by the d and α measurements.

downsampled by a factor of 4. The result of constraint line clustering with a 5 by 5 neighborhood is shown in Fig. 11. Since the motion data was downsampled by a factor of 4, the effective neighborhood size was 20 by 20. The original images were 256 by 256 pixels before filtering and computing the motion constraints. An idea of the relative size of the constraint line clustering neighborhood can be found by dividing the width of the original images by the effective width of the constraint line clustering neighborhood: the width of the neighborhood for constraint line clustering was almost 8 percent of the image width. Smoothed versions of the velocity field are provided in the experimental results in Section VI-D.

E. Summary

The constraint line clustering algorithm estimates the velocity vector in successive neighborhoods by intersecting the constraint lines in a neighborhood with the con-

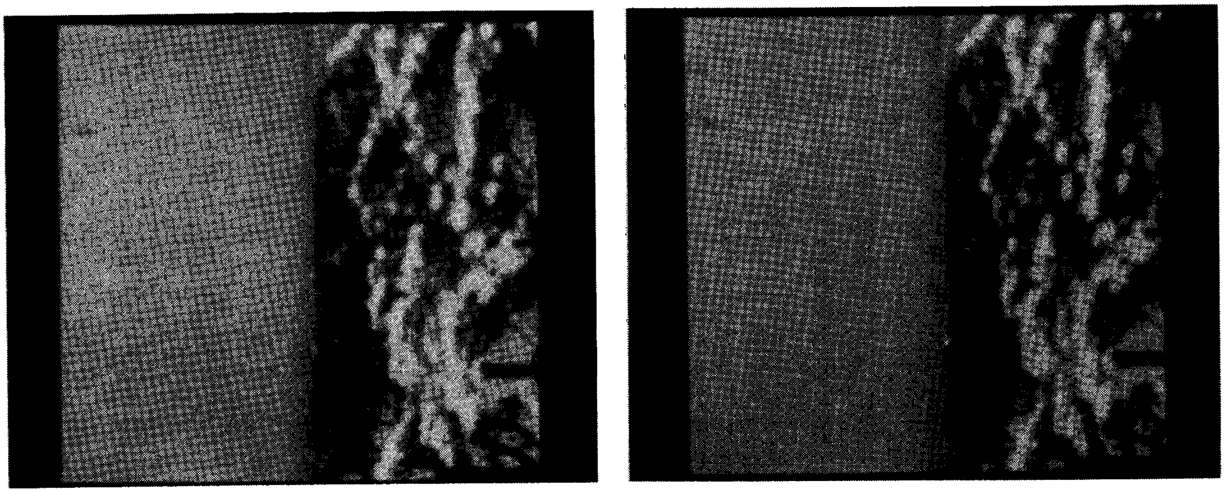

Fig. 7. A pair of real images that contains an object displaced slightly between frames. The scene contained a phonebook lying on a sheet of circuit board. The cover of the phonebook had an outdoor scene.

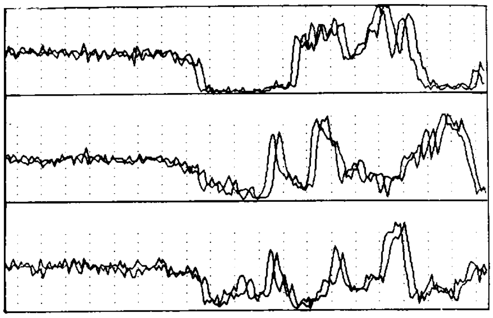

Fig. 8. Three pairs of horizontal slices through the images of the phonebook test case from Fig. 7 are plotted. The plots were created by overlaying the plots from the same position in the two frames of the image pair. The images are very noisy with considerable variation in the left hand portion of the plots which corresponds to the stationary part of the scene.

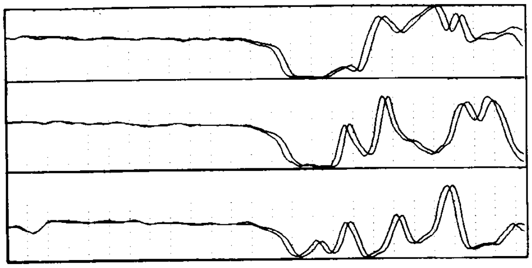

Fig. 9. Three pairs of horizontal slices through the images of the phonebook test case after smoothing with a 16 by 16 point Gaussian filter are plotted. The displacement between frames is clearly seen as the phase shift in the plots. This illustrates the degree of displacement appropriate to algorithms based on the image flow constraint equation.

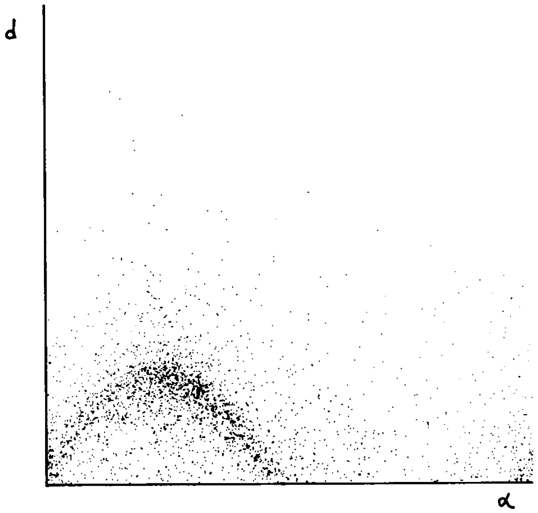

Fig. 10. A scatter plot of constraint line displacement d versus angle α for the motion constraints obtained from the phonebook test case is shown. This illustrates the poor quality of the motion data due to noise and the effects of occlusion.

straint line from the center of the neighborhood and choosing the midpoint of the tightest cluster. The tightest cluster is the interval of smallest width along the constraint line that contains half of the intersections. This test is motivated by the characteristics of motion boundaries. If the motion boundary is smooth, then at least half of the d and α measurements in the neighborhood belong to the same motion as the d and α measurement at the center of the neighborhood. It is harmless to assume the remaining measurements are not from the same region of motion; rejecting a helpful intersection is harmless, but accepting a bad one can ruin the motion estimate.

V. Error Analysis

This section provides some analysis of constraint line clustering that explains some of the characteristics of the algorithm that have been seen in experimental results. The analysis assumes that the distribution of constraint line orientations is uniform. This analysis may seem too op-

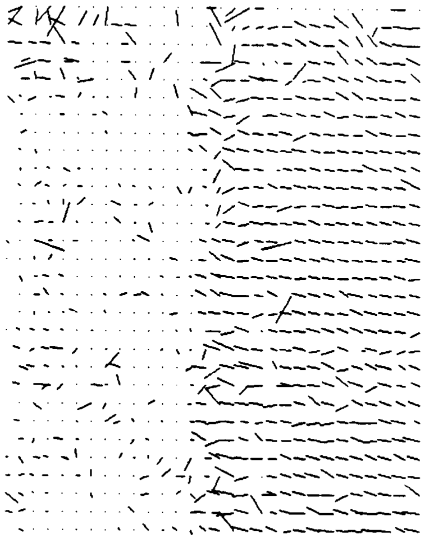

Fig. 11. The velocity field obtained by constraint line clustering with a 5 by 5 neighborhood is shown. The d and α arrays were downsampled by a factor of 4 to increase the effective neighborhood size to 20 by 20 without increasing computation.

timistic, but the neighborhood size for constraint line clustering can and should be large enough to provide sufficient variation in constraint line orientation for the assumption to be reasonable. Certainly the assumption is justified for the random dot images that are the key test case in this work. The important point addressed by the analysis is to offer an explanation of why constraint line clustering works in situations such as corners that violate the assumptions of the algorithm.

A. Constraint Line Intersections

The analysis begins by deriving an interesting formula for a geometric problem that will lead us to approximate formulas for the general case. Suppose that there are one or more lines intersecting a given line L and that these lines all pass through a point P, as depicted in Fig. 12. The line L is displaced from point P by distance d along the perpendicular bisector of L passing through P. The probability density function for the intersection positions will be determined. This problem corresponds to the situation where the one-dimensional cluster analysis algorithm is applied along a constraint line that is associated with a motion, but due to errors in the constraint line coefficients the line passes a distance d from the true velocity at point P and all constraint lines in the neighborhood of L are error free and pass through point P. This is an important case since experimental results indicate that the accuracy of constraint line clustering is determined predominantly by the accuracy of the estimate of the motion constraint coefficients. If the constraint line at the center of the neighborhood is accurate, but some of the constraint lines in the neighborhood are not, then constraint line clustering will filter out the inaccurate intersections. This analysis addresses the case where the constraint lines

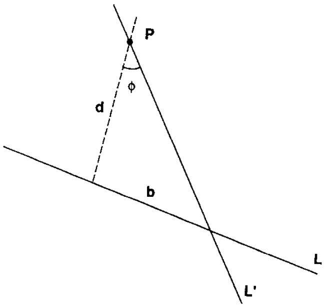

Fig. 12. A line L is displaced from a point P by distance d. Another line, passing through P with an angle ϕ between the line and the perpendicular bisector of L, intersects line L. The distance from the perpendicular bisector of L passing through P to the point of intersection is b. The probability distribution for b is derived in the text and is part of the error analysis of constraint line clustering.

surrounding the center are accurate, but the center constraint line is not.

The spread in intersection positions that will occur when multiple valid constraint lines intersect constraint line L must be determined. To obtain this statistic, the probability density function for the distance b along line L from the perpendicular bisector of L, passing through P, to the point of intersection must be determined. Assume that the angle between the valid constraint line and the perpendicular bisector of L is ϕ. The distance b is given by

$$b = d \tan \phi. \tag{17}$$

A probability density function for b can be derived from a probability density function for ϕ by equating equal probability measures and applying change of variables. For any value of ϕ_0 in the range $[-\pi/2, \pi/2]$,

$$\text{Prob}\,[\phi < \phi_0] = \int_{-\pi/2}^{\phi_0} p_\phi(\phi)\,d\phi = \text{Prob}\,[b < b_0] \tag{18}$$

where $b = d \tan \phi_0$, and

$$\text{Prob}\,[b < b_0] = \int_{-\infty}^{b_0} p_b(x)\,dx. \tag{19}$$

By change of variables

$$\int_{-\pi/2}^{\phi_0} p_\phi(\phi)\,d\phi = \int_{-\infty}^{b_0} p_\phi(\tan^{-1}(x/d))\frac{1}{1+(x/d)^2}\frac{1}{d}\,dx. \tag{20}$$

So the probability density function for b, in terms of the probability density function for ϕ is

$$p_b(x) = \frac{d}{d^2 + x^2} p_\phi(\tan^{-1}(x/d)). \tag{21}$$

To continue the analysis, assume that ϕ is uniformly distributed between $-\pi/2$ and $\pi/2$. This assumption should typically be valid since the size of a neighborhood should be large enough to guarantee that a diverse range of constraint line orientations lies in the neighborhood. The assumption of uniform distribution in ϕ leads to a Cauchy distribution for the intersection position:

$$p_b(x) = \frac{d/\pi}{x^2 + d^2}. \tag{22}$$

This distribution has an unbounded second moment, so the variance will not be a useful measure of spread. The percentage of the distribution that lies in the interval $[-l, +l]$ can be computed. The value of l that contains some fixed percentage (such as half) of the intersection positions can be used as a measure of the spread. The percentage of intersections in the interval $[-l, +l]$ is given by

$$\int_{-l}^{l} p_b(x)\, dx = 2 \int_0^l p_b(x)\, dx \tag{23}$$

$$= 2d/\pi \int_0^l \frac{1}{d^2 + x^2}\, dx \tag{24}$$

$$= \frac{2d}{\pi} \left[\frac{1}{d} \tan^{-1}(x/d) \right]_0^l \tag{25}$$

$$= 2/\pi \tan^{-1}(l/d). \tag{26}$$

The width s of the interval that contains a fraction P of the intersections is

$$s = 2d \tan(P\pi/2). \tag{27}$$

Note that the spread is proportional to d and the constant of proportionality is $2 \tan(P\pi/2)$. This constant is about 1 for P around 30 percent and is exactly 2 for $P = 50$ percent. Above 50 percent, the constant of proportionality increases rapidly.

B. Multiple Motions

When there are two motions near a constraint line, the intersections with the constraint line will have a distribution that is the sum of two Cauchy distributions, weighted according to the fraction of the intersections that arise from each motion. The combined distribution is

$$p_b(b \mid d_1, d_2) = \eta C(b \mid d_1) + (1 - \eta) C(b \mid d_2). \tag{28}$$

This result gives us an insight into the effect of interference between two motions. The spread of the intersections from each motion will be proportional to the distance of the motion from the constraint line. The constraint line clustering algorithm chooses the tightest cluster of intersections that contains roughly half of the intersections. This tends to guarantee that the cluster of intersections that corresponds to the majority of intersections (that is, the set of intersections from the same side of the motion boundary from which the constraint line was taken) will be chosen over the minority cluster. Severe interference can only take place if the majority cluster has a very large spread and overlaps the minority cluster sufficiently to add enough intersections to the minority cluster to make it seem to be the majority cluster. Since the distribution of intersections scales in proportion to the distance of the motion from the constraint line, this is an unlikely situation.

The assumptions used to formulate the test for the tightest cluster of intersections along a constraint are not valid at the corners of sharp motion boundaries, but the constraint line clustering algorithm works well at corners. This is explained by the statistical results presented above. At a corner, the statistics of constraint line intersections are given by (28). The intersections caused by constraint lines from inside the corner form a tight cluster with a Cauchy distribution while the intersections caused by constraint lines from outside the corner form a broad cluster with a Cauchy distribution. The broader Cauchy distribution trails off slowly from its central peak and approximates a uniform distribution over the interval that overlaps the tight cluster of the narrower Cauchy distribution. So in the vicinity of the small cluster of intersections from the constraint lines inside the corner there are additional intersections that are approximately uniformly distributed and these additional intersections do not bias the location of the tightest peak in the interval. The constraint line clustering algorithm will, under the most probable circumstances, choose the location of the center of the small, tight cluster of intersections from the constraint lines inside the corner as the motion estimate over some other location near the much broader spread of intersections from the constraint lines outside the corner.

VI. Smoothing Discontinuous Image Flows

An algorithm called surface-based smoothing for improving the velocity field estimate of an image flow vector field is described in this section. The algorithm iteratively computes the motion boundaries in the velocity field and smooths the velocity field estimate between the motion boundaries. An excellent velocity field estimate with sharp motion boundaries can be derived from a noisy initial velocity field estimate. The algorithm is based on a fundamental property of all intrinsic images and can be used to improve the estimate of any intrinsic image.

Surface-based smoothing is able to improve the velocity estimate produced by constraint line clustering because the velocity estimate is good enough to allow the motion boundaries to be detected, so smoothing can be restricted to regions between motion boundaries. The algorithm relies on the fact that the true velocity function is smooth between motion boundaries because the moving surfaces in the scene that generated the image flow field are smooth between step changes in depth.

A. Motion Boundary Detection

The velocity field estimated by constraint line clustering is good enough to allow any of the usual edge detection algorithms to be used to detect the motion boundary, but better performance can be obtained by using a better edge detection method. Thompson, Mutch and Berzins [34] presented an algorithm for motion boundary detection that was an extension of the edge detection algorithm of Marr and Hildreth [30] and presented insights on occlusion analysis. In this work, the gradient of the Gaussian followed by nonmaxima suppression is used to detect boundaries in the components of the image flow field [46].

After each velocity field component is smoothed with a Gaussian filter, the Prewitt operator is used to compute the x and y partial derivatives of each motion estimate [47, p. 106]. The Prewitt masks are applied to the smoothed u and v motion data to produce four arrays of boundary data: the x and y partial derivative approximations for the each of the two velocity field components. The partial derivative approximations are used to compute the gradient magnitude and angle for each velocity field component.

Nonmaxima suppression reduces the broad ridges and isolated mounds of edge detector output to thin curves and edge fragments. The gradient orientation is reduced to one of four sectors, then the magnitude of the gradient at each point is compared to the magnitude of its two neighbors along the line of the sector. If either of the neighbors of the center element have large magnitudes, then the magnitude of the center element is set to zero. The total number of edge detector output values that are above some threshold is decreased. This makes more of the velocity field data available to the smoothing algorithm. Nonmaxima suppression also produces paths through weak boundaries that allow the velocity field estimates on opposite sides of the boundaries to mix.

B. Velocity Field Smoothing

The smoothing problem is formulated as a minimization problem with two components: 1) a cost for the lack of smoothness in the velocity field between motion boundaries and 2) a cost on the deviation of the smoothed velocity field from the velocity field estimated by constraint line clustering. This approach was motivated by the work of Horn and Schunck [9] and Grimson [48]–[50]. Other methods for implementing the smoothing equations are discussed by Terzopoulos [51] and an interesting discussion of viewpoint invariant surface interpolation is provided by Blake [52]. The mathematics of smoothing operators is presented by Brady and Horn [53]. Surface interpolation at multiple scales of resolution is thoroughly discussed by Terzopoulos [51], [54]. Markov random field methods for surface reconstruction have been discussed by Marroquin [55] and Geman and Geman [56]. The relationship of problems in surface reconstruction to regularization has been described by Poggio and Torre [57]. An analog network model for the implementation of surface reconstruction algorithms based on Markov random fields was presented by Hutchinson and Koch [58]. Blake and Zisserman [59] present a thorough discussion of algorithms for the reconstruction of surfaces in vision.

Let u and v denote the x and y components of the velocity field estimate obtained by constraint line clustering, let $\bar{u}$ and $\bar{v}$ denote the smoothed velocity field estimate. The cost assigned to the deviation of the smoothed velocity field estimate from the original velocity field estimate is

$$(u - \bar{u})^2 + (v - \bar{v})^2. \qquad (29)$$

The cost assigned to lack of smoothness is the sum of the squares of the magnitudes of the gradients of the x and y velocity field components,

$$\bar{u}_x^2 + \bar{u}_y^2 + \bar{v}_x^2 + \bar{v}_y^2, \qquad (30)$$

as used by Horn and Schunck [9]. The combined optimization criterion is

$$\int\int \kappa^2(\bar{u}_x^2 + \bar{u}_y^2 + \bar{v}_x^2 + \bar{v}_y^2) + (u - \bar{u})^2 + (v - \bar{v})^2 \, dx \, dy \qquad (31)$$

where κ^2 weights the degree of smoothing versus the degree of deviation from the original velocity field estimate. This optimization problem is easily solved by the calculus of variations [60, pp. 191–192]. The result is a pair of linear partial differential equations that are decoupled:

$$\kappa^2\nabla^2\bar{u} - u = 0 \qquad (32)$$

$$\kappa^2\nabla^2\bar{v} - v = 0. \qquad (33)$$

The equations can be solved iteratively by approximating the Laplacian with a nine point mask [61, p. 128],

$$\nabla^2 \approx \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}. \qquad (34)$$

Using this approximation, the iterative equations are easily obtained

$$\bar{u}^{n+1} = \frac{1}{1 + 20\kappa^2}\left[u^n + \kappa^2 \mathcal{S}(\bar{u}^n)\right] \qquad (35)$$

$$\bar{v}^{n+1} = \frac{1}{1 + 20\kappa^2}\left[v^n + \kappa^2 \mathcal{S}(\bar{v}^n)\right], \qquad (36)$$

where the function $\mathcal{S}(\cdot)$ represents the computation of the eight point surround of the approximation to the Laplacian.

These smoothing equations should only be used over the interior of a region of slowly varying velocity. The equations should never be applied across a motion boundary since to do so would blur the boundary. Natural boundary conditions should be used when the support of the approximation to the Laplacian is next to a motion boundary. Since the shape of the motion boundary is arbitrary, it is difficult to develop and apply the approximations to the Laplacian that should be used at a motion boundary. A simple scheme was used to approximate natural boundary conditions near a motion boundary: the velocity field value at the center of the Laplacian is used in place of the velocity field value at any point in the support of the Laplacian that is a boundary point. This scheme is simple to implement and works for boundaries of arbitrary shape.

C. Interleaved Detection and Smoothing

The algorithm for detecting motion boundaries and the algorithm for smoothing the velocity field estimate between motion boundaries are interleaved so that each computation can benefit from the improved information provided by the other. The computation begins with the original velocity field estimate computed by constraint line clustering. The x and y components of the velocity field are contained in separate arrays. The motion edge detector is applied to each velocity component to obtain separate arrays of edge information for each velocity component. One iteration of the smoothing operator is applied in place to each component of the velocity field, using the boundary information for that component, to obtain the new smoothed version of each velocity field component. The border of the arrays is filled using data from the rows and columns just inside the border to approximate natural boundary conditions along the perimeter of the field of view. Subsequent iterations begin by applying the motion boundary detection operator to the smoothed velocity field estimate. The smoothing that was performed between motion boundaries over the interior of surfaces will have reduced the variation in the velocity field. Since the variation is less, the output of the boundary detection operator in these regions will be weaker. Near a motion boundary, the local variation in the velocity field will be more apparent: the velocity values on either side of the boundary will be more consistently the same and the velocity values on opposite sides of the boundary will be more consistently different. The velocity values in the interior of the surfaces corresponding to objects and the background will not wander far from the original values estimated by constraint line clustering since there is a cost on the deviation of the smoothed velocity estimates from the original velocity estimates. This anchors the velocity estimates to the original motion data. As successive iterations are performed, the height of the boundary operator output will increase along a motion boundary, decrease over the interior of objects and background points between boundaries, and the smoothing of the velocity estimate will be confined to a well within which a smooth velocity field estimate will be achieved independent of the velocity field values at points on the other side of the motion boundary.

D. Experimental Results

The boundary estimation and surface smoothing computation was applied to the results of constraint line clustering for the random box test case shown in Fig. 4. The results of surface-based smoothing are displayed in Figs. 13–15. The plot in Fig. 15 shows that the maximum deflection of the estimated velocity field from the true velocity field is around 3 percent. Maximum deflection is a better statistic than mean-squared error since extreme errors are significant. The results show that the interleaved iterations of boundary estimation and surface smoothing can improve the velocity field estimate and produce a clean estimate of the motion boundary.

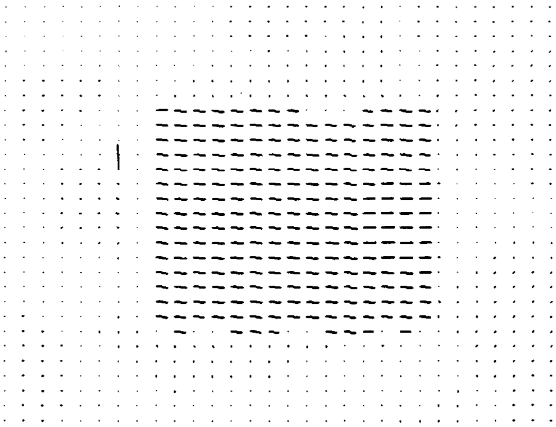

Fig. 13. The velocity field computed after 16 smoothing iterations have been performed on the velocity field shown in Fig. 4 is displayed. The motion edges were computed using the Prewitt x and y partial derivative approximations applied to the motion components after smoothing with an 8 by 8 Gaussian filter. The motion edge threshold was 1000 and 16.0 was used for the smoothing weight.

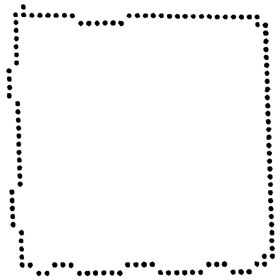

Fig. 14. The plot is the boundary of the u component of the velocity field shown in Fig. 13. Note that there are no false boundary points.

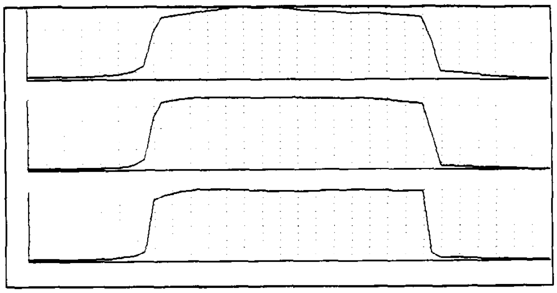

Fig. 15. Three horizontal slices through the u component of the smoothed velocity field from Fig. 13 are graphed. The deflection of the estimated value from the true value is no greater than 3 percent.

Surface-based smoothing was also applied to the output of constraint line clustering for the phonebook example shown in Fig. 11. The results of surface-based smoothing for the phonebook test case are shown in Figs. 16 and 17.

VII. Summary and Conclusions

This paper presented work on velocity field estimation with emphasis on image flows that may contain step discontinuities in the image irradiance or velocity field. A

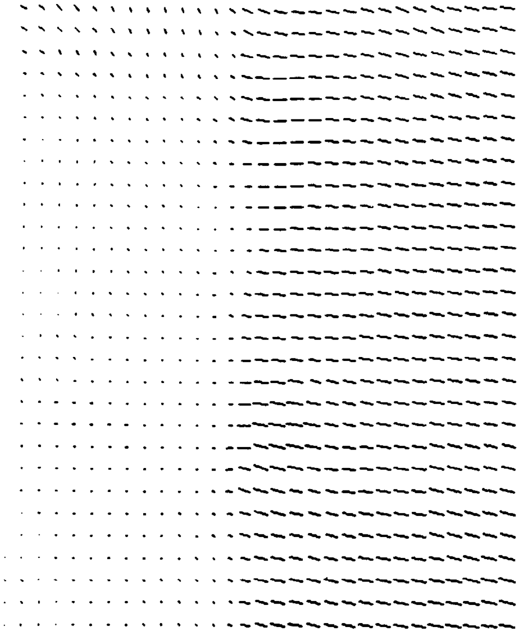

Fig. 16. The smoothed velocity field after 16 iterations of surface-based smoothing applied to the output of constraint line clustering from Fig. 11 for the phonebook example is shown. The smoothing parameter was 16.0 and the threshold on the motion edge operator output was 2500. The u and v velocity components had to be scaled down by a factor of 5 to prevent integer overflow in the iterative calculations.

one-dimensional cluster analysis algorithm for the estimation of discontinuous image flows was described and a partial analysis of errors was presented. The paper concluded with a presentation of a smoothing algorithm that improves velocity field estimates without blurring the motion boundaries.

Two contributions important to vision research were presented: 1) the constraint line clustering algorithm that uses a clever statistical test to estimate the image flow velocity field in the presence of step discontinuities and 2) the surface-based smoothing algorithm that improves velocity field estimates without blurring motion boundaries. The constraint line clustering algorithm shows that it is possible to formulate an algorithm with statistical techniques that estimates an intrinsic image without losing the boundary information that is important to perception. The surface-based smoothing algorithm shows that an improved approximation to an intrinsic image can be obtained without blurring boundaries in the estimate of the intrinsic image. These two results have implications for vision research beyond image flow estimation.

Acknowledgment

Many thanks to my thesis committee for their support during this work, especially B. Horn and M. Brady for their immense technical skill and patience; R. Jain for his encouragement and his efforts to provide an excellent research environment in the Artificial Intelligence Laboratory at the University of Michigan; and the reviewers for their many helpful comments, especially the references to related work in robust statistics.

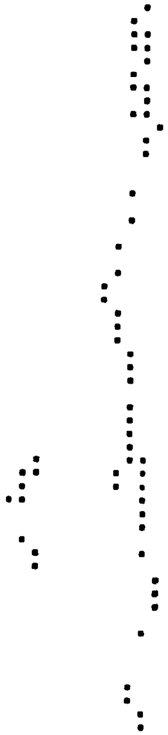

Fig. 17. The boundary in the u component of the velocity field for the phonebook example after 16 iterations is shown. The boundary is not dense due to poor choice of threshold, but the boundary estimate is good enough for surface-based smoothing without excessive leakage of motion data from one region to another as shown in the smoothed velocity field in Fig. 16.

References

[1] B. G. Schunck, "The image flow constraint equation," *Comp. Vision, Graph. Image Proc.*, vol. 35, pp. 20–46, 1986.

[2] ——, "The motion constraint equation for optical flow," in *Proc. Int. Joint Conf. Pattern Recogn.*, 1984, pp. 20–22.

[3] ——, "Motion segmentation and estimation," Ph.D. thesis, Dep. Elec. Eng. Comput. Sci., M.I.T., Cambridge, MA, 1983.

[4] ——, "Image flow continuity equations for motion and density," in *Proc. Workshop Motion*, 1986.

[5] A. Verri and T. Poggio, "Qualitative information in the optical flow," in *Proc. DARPA Image Understanding Workshop*, 1987, pp. 825–834.

[6] O. Braddick, "A short-range process in apparent motion," *Vision Res.*, vol. 14, pp. 519–527, 1974.

[7] H. Nagel, "From digital picture processing to image analysis," in *Proc. Int. Conf. Image. Anal. Processing*, 1980.

[8] T. Poggio, "Early vision: From computational structure to algorithms and parallel hardware," *Comp. Vision, Graph. Image Proc.*, vol. 31, pp. 139–155, 1985.

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

[10] T. Poggio and W. Reichardt, "Visual control of orientation behavior in the fly," *Quart. Rev. Biophys.*, vol. 9, pp. 377–438, 1976.

[11] W. Reichardt and T. Poggio, "Figure-ground discrimination by relative movement in the visual system of the fly," *Bio. Cybern.*, vol. 35, pp. 81–100, 1979.

[12] E. Buckner, "Elementary movement detectors in an insect visual system," *Bio. Cybern.*, vol. 24, pp. 85–101, 1976.

[13] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems.* New York: Wiley, 1980.

[14] B. G. Haskell, "Frame-to-frame coding of television pictures using two-dimensional Fourier transforms," *IEEE Trans. Inform. Theory*, vol. 20, pp. 119-120, 1974.

[15] J. O. Limb and J. A. Murphy, "Measuring the speed of moving objects from television signals," *IEEE Trans. Commun.*, vol. 23, pp. 474-478, 1975.

[16] ——, "Estimating the velocity of moving images in television signals," *Comp. Graph. Image Proc.*, vol. 4, pp. 311-327, 1975.

[17] C. Cafforio and F. Rocca, "Methods for measuring small displacements of television images," *IEEE Trans. Inform. Theory*, vol. 22, pp. 573-79, 1976.

[18] A. N. Netravali and J. D. Robbins, "Motion-compensated television coding: Part I," *Bell Syst. Tech. J.*, vol. 58, pp. 631-670, 1979.

[19] J. A. Stuller and A. N. Netravali, "Transform domain motion estimation," *Bell Syst. Tech. J.*, vol. 58, pp. 1673-1702, 1979.

[20] J. A. Stuller, A. N. Netravali, and J. D. Robbins, "Interframe television coding using gain and displacement compensation," *Bell Syst. Tech. J.*, vol. 59, pp. 1227-1240, 1980.

[21] R. A. Jones and H. Rashid, "Residual recursive displacement estimation," in *Proc. Conf. Pattern Recog. Image Processing*, 1981, pp. 508-509.

[22] R. Paquin and E. Dubois, "A spatio-temporal gradient method for estimating the displacement field in time-varying imagery," *Comp. Vision, Graph. Image Proc.*, vol. 21, pp. 205-221, 1983.

[23] L. S. Davis, Z. Wu, and H. Sun, "Contour-based motion estimation," Tech. Rep., Comput. Vision Lab., Univ. Maryland, 1981.

[24] E. C. Hildreth, "The computation of the velocity field," *Proc. Roy. Soc. Lond. B*, vol. 221, pp. 189-220, 1984.

[25] E. C. Hildreth, *The Measurement of Visual Motion.* Cambridge, MA: M.I.T. Press, 1984.

[26] E. C. Hildreth and S. Ullman, "The measurement of visual motion," Memo 699, Artificial Intelligence Lab., M.I.T., Cambridge, MA, 1982.

[27] S. M. Anstis, "Phi movement as a subtraction process," *Vision Res.*, vol. 10, pp. 1411-1430, 1970.

[28] S. M. Anstis and B. J. Rogers, "Illusory reversal of visual depth and movement during changes of contrast," *Vision Res.*, vol. 15, pp. 957-961, 1975.

[29] S. Ullman, *The Interpretation of Visual Motion.* Cambridge, MA: M.I.T. Press, 1979.

[30] D. Marr and E. C. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. Lond. B*, vol. 207, pp. 187-217, 1980.

[31] K. Nakayama and J. Loomis, "Optical velocity patterns, velocity-sensitive neurons, and space perception: A hypothesis," *Perception*, vol. 3, pp. 63-80, 1974.

[32] J. Batali and S. Ullman, "Motion detection and analysis," in *Proc. DARPA Image Understand. Workshop*, Nov. 1979.

[33] K. M. Mutch and W. B. Thompson, "Analysis of accretion and deletion at boundaries in dynamic scenes," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 7, pp. 133-138, Mar. 1985.

[34] W. B. Thompson, K. M. Mutch, and V. A. Berzins, "Dynamic occlusion analysis in optical flow fields," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 7, pp. 374-383, July 1985.

[35] C. L. Fennema and W. B. Thompson, "Velocity determination in scenes containing several moving objects," *Comp. Graph Image Proc.*, vol. 9, pp. 301-315, 1979.

[36] B. G. Schunck, "Motion segmentation and estimation by constraint line clustering," in *Proc. Workshop Comput. Vision*, 1984, pp. 58-62.

[37] D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing," *Proc. Roy. Soc. Lond. B*, vol. 211, pp. 151-180, 1981.

[38] B. Noble and J. W. Daniel, *Applied Linear Algebra.* Englewood Cliffs, NJ: Prentice-Hall, 1977, 2nd ed.

[39] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications.* New York: Wiley-Interscience, 1974.

[40] T. L. Boullion and P. L. Odell, *Generalized Inverse Matrices.* New York: Wiley-Interscience, 1971.

[41] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems.* Englewood Cliffs, NJ: Prentice-Hall, 1974.

[42] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. Assoc. Comp. Mach.*, vol. 24, pp. 381-395, June 1981.

[43] H. A. David, *Order Statistics.* New York: Wiley, 1981, 2nd ed.

[44] P. J. Huber, *Robust Statistics.* New York: Wiley, 1981.

[45] R. J. Serfling, *Approximation Theorems of Mathematical Statistics.* New York: Wiley, 1980.

[46] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 8, pp. 679-698, 1986.

[47] R. Nevatia, *Machine Perception.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[48] W. E. L. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Vision System.* Cambridge, MA: M.I.T. Press, 1981.

[49] W. E. L. Grimson, "A computational theory of visual surface interpolation," Memo 613, Artificial Intelligence Lab., M.I.T., 1981.

[50] W. E. L. Grimson, "The implicit constraints of the primal sketch," Memo 663, Artificial Intelligence Lab., M.I.T., 1981.

[51] D. Terzopoulos, "Multilevel reconstruction of visual surfaces: Variational principles and finite element representations," Memo 671, Artificial Intelligence Lab., M.I.T., 1982.

[52] A. Blake, "Reconstructing a visible surface," in *Proc. Nat. Conf. Artificial Intelligence*, 1984, pp. 23-26.

[53] M. Brady and B. K. P. Horn, "Rotationally symmetric operators for surface interpolation," Memo 654, Artificial Intelligence Lab., M.I.T., Nov. 1981.

[54] D. Terzopoulos, "Multilevel computational processes for visual surface reconstruction," *Comp. Vision, Graph. Image Proc.*, vol. 24, pp. 52-96, 1983.

[55] J. L. Marroquin, "Surface reconstruction preserving discontinuities," Memo 792, Artificial Intelligence Lab., M.I.T., 1984.

[56] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 6, pp. 721-741, 1984.

[57] T. Poggio and V. Torre, "Ill-posed problems and regularization analysis in early vision," Memo 773, Artificial Intelligence Lab., M.I.T., Apr. 1984.

[58] J. M. Hutchinson and C. Koch, "Simple analog and hybrid networks for surface interpolation," in *Neural Networks for Computing*, J. S. Denker, Ed. New York: Amer. Inst. Phys., 1986, pp. 235-240.

[59] A. Blake and A. Zisserman, *Visual Reconstruction.* Cambridge, MA: M.I.T. Press, 1987.

[60] R. Courant and D. Hilbert, *Methods of Mathematical Physics.* New York: Wiley, 1937, vol. 1.

[61] W. F. Ames, *Numerical Methods for Partial Differential Equations.* New York: Academic, 1977, 2nd ed.

Brian G. Schunck (S'75-M'83) has worked for several years on the development of systems for machine vision and industrial automation. He was educated in computer science at the University of California, Irvine, where he received the B.S. magna cum laude in 1976. He studied electrical engineering, systems theory, and artificial intelligence at M.I.T. where he received the Master's and E.E. degrees in 1979 for work on the control of robotic manipulators and the doctorate in 1983 for research on image flow.

He has worked for several companies on projects in machine vision, robotics, control systems and electronics, and computer architecture. He is currently an Assistant Professor in the Department of Electrical Engineering and Computer Science and a member of the Artificial Intelligence Laboratory at the University of Michigan, Ann Arbor. His interests include statistical methods for machine vision and industrial inspection, qualitative reasoning and constraint propagation in engineering design, electronic media, and marine navigation.