

Motion-Model-Based Boundary Extraction

Hongche Liu^{‡†}, Tsai-Hong Hong[‡], Martin Herman[‡], and Rama Chellappa[†]
liu@cme.nist.gov, hongt@cme.nist.gov, herman@cme.nist.gov, chella@eng.umd.edu

[‡]Intelligent Systems Division, National Institute of Standards and Technology (NIST)
Bldg. 220, Rm B124, Gaithersburg, MD 20899

[†]Center for Automation Research/Department of Electrical Engineering,
University of Maryland, College Park 20742

Abstract

Motion boundary extraction and optical flow computation are two subproblems of the motion recovery problem that cannot be solved independently of each other. They represent the most common dilemma in motion research. A popular approach uses an iterative scheme that consists of motion boundary extraction and optical flow computation components and refines each result through iteration. This approach is typically time-consuming and sometimes does not converge. We present a local, non-iterative algorithm that extracts motion boundaries and computes optical flow simultaneously. This is achieved by modeling a 3-D image intensity block with a general motion model that presumes locally coherent motion. Local motion coherence, which is measured by the fitness of the motion model, is the criterion we use to determine whether motion should be estimated, or otherwise motion boundaries should be located. The motion boundary extraction algorithm is evaluated quantitatively and qualitatively against other existing algorithms in a scheme originally developed for edge detection. The results show that our algorithm is accurate in locating boundaries. The flow portion of the algorithm is presented in another paper[24].

Keywords: motion analysis, segmentation.

1. Introduction

This paper studies the strengths and weaknesses of recent motion boundary detection and motion segmentation algorithms and proposes a local, non-iterative algorithm for motion boundary detection with potential for real-time implementation. This algorithm extracts motion boundaries and computes optical flow at the same time. We apply a quantitative evaluation scheme for boundary detection to show that our algorithm is accurate in locating motion boundaries.

In this paper, the problem of *motion recovery* is referred to as involving two major subproblems: *optical flow computation* and *motion segmentation*. Optical flow computation quantitatively measures the motion associated with the perceived objects; motion segmentation, on the other hand, qualitatively distinguishes different moving objects. The fact that they are dependent on each other has complicated the general motion recovery problem.

Due to the aperture problem, early motion estimation algorithms [17] [18] usually enforced a smooth flow field as an additional constraint. Recent approaches use spatio-temporal filters [10] [14] [24], often with large support, to estimate image properties and then solve for optical flow. In either case, on or near motion boundaries, this smoothing or filtering renders the estimation incorrect. In other words, motion estimation is not accurate until we know where the boundaries are. On the other hand, motion boundaries are defined as motion field discontinuities. (The motion field is qualitatively equivalent to the optical flow field [34].) Due to the aforementioned optical flow error around motion boundaries, the requirement of a dense flow field, and noise in the optical flow field, the motion boundaries are very difficult to extract and/or locate from optical flow. Researchers have used other image cues, for example, accretion and deletion [26], or normal flow [17], to detect motion discontinuities, but they are not always correct for all situations because they provide only partial information about the motion. In other words, motion boundaries cannot be located accurately without a dense and accurate optical flow field.

Even though they are two aspects of a single problem, optical flow computation has received much more attention in the literature than motion boundary extraction. Existing methods for

motion boundary extraction are approached through optical flow algorithms. A popular technique is to use an iterative scheme that consists of two components: optical flow estimation and motion boundary extraction. The basic idea is to refine both components' results through iteration. This approach is time-consuming and sometimes does not converge. We believe that optical flow and motion boundaries are of equal importance and we present an algorithm that produces both outputs at the same time.

Although “global” motion segmentation may be more convenient for other motion applications, we realize, from the above analysis, that “local” motion boundary detection/extraction is sufficient for combining with optical flow an algorithm for motion recovery. In fact, our view is that local image properties provide abundant information and motion estimation should be performed pointwise [23].

The local properties that we use are image spatial and temporal derivatives up to third order. It has been shown in [23] that Hermite polynomial differentiation filters are very stable and insensitive to noise even up to this high order. With the aid of sufficient and accurate image properties, a motion-model-based approach to boundary extraction becomes possible. In this approach, we fit the image properties with a single coherent motion model, which leads to a linear system of multiple motion constraint equations. Pixels that fit the model are locations where the motion is coherent, so the motion can be estimated using the linear system. Those pixels that do not fit the model represent failures of the motion model in describing the local motion. A failure of the model can only be attributed to multiple motions existing in the local window used to estimate the image properties, assuming that brightness constancy is maintained. Using a least square error method on the overdetermined linear system, a failure of the model is measured by the residual. An analysis of the residual is shown to reflect the likelihood of a motion boundary. Using an optimal filter on the residual, we can easily locate motion boundaries.

Using the residual for motion boundary extraction offers several advantages over using flow. First, the residual is a scalar, so it avoids the difficulty of handling vector field discontinuities while providing equivalent information about motion boundaries, e.g., whenever one component

of the flow is discontinuous, the residual is high. Second, flow values on the boundaries are not accurate and are very noisy, and thus require smoothing for boundary extraction. This extra smoothing may cause localization error. Third, the residual is computed using a 3-D motion model so that it corresponds to real motion boundaries and it is not susceptible to nonuniform flow within an object; whereas nonuniform flow can induce false detections of discontinuities in flow-based methods.

The appeal of a local, non-iterative approach lies in its potential speed. However, its accuracy should not be compromised. To measure the accuracy, we need an evaluation scheme to compare different motion boundary extraction algorithms. Since recent approaches combine optical flow and motion boundary detection, evaluation has often been performed based on the final optical flow. This has the disadvantage of not distinguishing the source of error, which may be due to inaccurate optical flow or inaccurate motion boundary location. In other words, evaluation based on segmented optical flow does not suggest a direction for improvement. Hence, we employ here a quantitative evaluation scheme applied only to motion boundary extraction. This scheme takes into account not only the probabilities of detection and miss but also localization error. This scheme was originally developed for edge detection [16] .

The rest of the paper is organized as follows. Section 2 surveys previous work and its strengths and weaknesses. The motion model and our algorithm are introduced in Section 3. Discussions of motion boundary properties are presented in Section 4. The evaluation scheme and experimental results based on it are presented in Section 5. Section 6 concludes the paper with a statement of our contributions and future prospects for this work.

2. Previous Work

Braddick’s psychological experiments on random dot motion [5] set the stage for vision research on motion boundaries. It verified the human visual capability of perceiving motion boundaries clearly without any other visual cues such as texture. Table 1 summarizes the existing work on motion boundary extraction or segmentation. This survey is not exhaustive but represents

typical work in this area, which will be elaborated in the following subsections.

Table 1: Summary of current motion boundary extraction algorithms

Non-iterative schemes		Iterative schemes	
Motion boundary extraction vs. flow estimation	Algorithm by	Techniques	Algorithm by
Prior to	Hildreth [17]	Pyramid linking	Hartley [12]
	Spoerri & Ullman [32]	Markov random field with binary line processes	Koch, Marroquin & Yuille [21]
Simultaneous with	Mutch & Thompson [26]		Murray & Buxton [25]
	Schunck [30]		Heitz & Bouthemy [15]
	Shizawa & Mase [31]	Tracking & nulling	Bergen et al. [4]
After	Potter [29] , Nakayama & Loomis [27] , Adiv [2] , Thompson, Mutch, & Berzins [33] , Dengler [9]	Robust estimation	Darrel & Pentland [7] Jepson & Black [19]

2.1 Non-iterative algorithms

Early research on motion boundary extraction or motion segmentation can be roughly characterized as based on a non-iterative approach. These algorithms can also be put into three categories [9] [32] based on whether the motion boundary extraction is performed prior to, simultaneously with, or after the flow field estimation (Refer to Table 1).

The approaches that extract motion boundaries prior to flow field estimation employ “motion primitives”[32] , usually normal flow [17] , as a basis. Hildreth’s method [17] is based on the intensity zero-crossing contours, which are different from motion contours, and may well cross motion boundaries. Traveling along a contour, the algorithm detects a boundary point as the sign of the normal flow changes. The method has two limitations [17] : first, it does not detect boundaries when the neighboring moving objects are traveling in about the same direction; second, it requires that there be two edge points with the same orientation in the contour. In addition, due to the use of contours, the boundaries detected are sparse, which is very restrictive for general applications. Spoerri & Ullman [32] use a local histogram on motion primitives and statistical tests to

infer motion boundaries, or tracking of “thin-bars” to find occlusion. This method is quite appealing for the diversity of statistical tests offered, but the motion primitives do not always provide sufficient information about the boundaries. The experiments show detection capabilities but the localization errors are significant even in synthetic images.

The approaches that extract motion boundaries simultaneously with flow field estimation include [26] ,[30] ,and [31] . Mutch & Thompson [26] use the fact that motion around occlusion boundaries induces accretion and deletion so that a local no-match between successive frames can signal occlusion boundaries, whereas a match can be used for estimating optical flow. This algorithm detects only occlusion boundaries but not all motion boundaries, for instance, two neighboring objects moving in parallel directions or a rotating object where no accretion or deletion occurs, will not be detected. Schunck’s algorithm [30] pays special attention to avoiding optical flow ambiguity at motion boundaries by constraint line clustering. The motion boundaries are actually detected from the flow field discontinuities. We categorize this algorithm as performing simultaneous estimation and segmentation because of its special treatment in handling boundary flow. The basic idea of the algorithm is to use local consensus to assign flow values instead of smoothing. The algorithm produces high localization error on motion boundary “corners”. The clustering technique is heuristic and is prone to numerical instability. Shizawa & Mase [31] use “multiple-flow constraint equations”, a generalization of the common optical flow constraint equation, to deal with motion boundaries and/or transparent motion. The algorithm generates not only flow but also a measure of the degree of multiplicity of motion. When a pixel’s associated multiplicity is determined to be greater than one, it is the motion boundary or where transparent motion occurs. This method unifies rather than distinguishes motion boundaries and transparent motion. However, it is more suitable for transparent motion than motion boundary extraction because of the assumption of additive multiple flows, which is less valid around motion boundaries.

The approach that extracts motion boundaries after flow field estimation is the most popular one. Global techniques such as the Hough transform (Adiv [2]), region growing (Potter [29]),

and pyramid linking (Dengler [9]) have been proposed. Local techniques include center-surround filters (Nakayama & Loomis [27]), and direction reversals of the Laplacian operator on the flow vector field (Thompson, Mutch, & Berzins [33]), etc. This approach offers only a partial solution to the motion estimation problem because the boundary depends heavily on the accuracy of the optical flow. However, this approach offers an algorithm suitable for one component of an iterative scheme. Such schemes are described next.

2.2 Iterative approach

The iterative method of motion estimation is an approach developed more recently. It has both optical flow estimation and segmentation components. These components interact with each other and improve their individual results during the course of the iteration. Pyramid linking, Markov random fields with line processes, robust estimation, and tracking plus nulling techniques have been proposed. Iterative methods tend to be more accurate than non-iterative methods but are time-consuming. Note that there are algorithms that use an iterative scheme to compute optical flow only. However, we do not label them as iterative methods here since they do not include the segmentation component.

Hartley's algorithm [12] uses an iterative pyramid linking technique for flow field segmentation. Segmentation is done by hierarchical linking and the flow field is computed and smoothed by fitting a linear or quadratic flow field model to the current flow. The algorithm is efficient and always converges but its overall accuracy depends heavily on the initial flow values, which the author does not address.

The use of a Markov random field model for flow has been proposed by Koch, Marroquin & Yuille [21] , Murray & Buxton [25] , and Heitz & Bouthemy [15] . They handle flow discontinuities by introducing a binary line process to discourage smoothing across boundaries. Although the reason for modeling the flow fields as Markov random fields is not clear, the results of these algorithms are generally good. The computational cost, however, is formidable (usually hundreds of iterations, or image sweeps).

Robust estimation techniques have been proposed by Darrel & Pentland [7] , and Jepson & Black [19] . They use a multi-layered motion model (“mixture model”[19]) and thus are capable of handling loosely occluded scenes (e.g. tree leaves) or transparent motion. The main idea is to estimate the dominant motion(s) in a window while rejecting inconsistent constraints as outliers so as to minimize their influence on the results. The results are promising when the algorithms converge.

Instead of using a layered motion model, Bergen et al. [4] model the addition of motions of differently moving image patterns (not necessarily square, as dictated by the window). A simple tracking and a ‘nulling’ mechanism is used to separate and estimate individual motions. In other words, a image registration and residual motion estimation are iterated. This algorithm has potential for high speed implementation on a system with warping hardware. The results are reasonably good but the algorithm may not always converge, depending on the noise level.

The results of the iterative methods seem good, but they have two major problems. The first is the computational load. the second is that the convergence rate depends on the scene, noise, and motion. Moreover, some of these algorithms may not converge at all.

3. Motion-Model-Based Boundary Extraction

The basic idea of our motion-model-based boundary extraction method is to fit the local image properties with a general motion model. The necessary elements of the scheme are a general motion model which is based on arbitrary 3-D motion; an accurate estimate of image properties, for which we use image spatial and temporal derivatives; and a procedure to measure the goodness of the image properties to the motion model. The following subsections briefly present the derivations of these three elements; the details can be found in [23] and [24] .

3.1 The general motion model

Here we describe an image motion model and an image motion equation that relates the spatio-temporal derivatives of the image intensity patterns in a sequence. Let a 3-D point

$\dot{P} = (X, Y, Z)^{T*}$ in the scene undergo steady small rotation $(\Omega_X, \Omega_Y, \Omega_Z)$ and translation (T_X, T_Y, T_Z) per unit time. Using the 3-D motion transformation matrix M

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \text{ where } M = \begin{bmatrix} 1 & -\Omega_Z & \Omega_Y & T_X \\ \Omega_Z & 1 & -\Omega_X & T_Y \\ -\Omega_Y & \Omega_X & 1 & T_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

$$\text{and } \begin{bmatrix} \dot{P}(t) \\ 1 \end{bmatrix} = M^t \begin{bmatrix} \dot{P}(0) \\ 1 \end{bmatrix}, \quad (2)$$

$$\text{we derive } \begin{bmatrix} \dot{P}(t) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -t\Omega_Z & t\Omega_Y & ta_X \\ t\Omega_Z & 1 & -t\Omega_X & ta_Y \\ -t\Omega_Y & t\Omega_X & 1 & ta_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3)$$

where each of a_X, a_Y, a_Z is a function of all of $(\Omega_X, \Omega_Y, \Omega_Z, T_X, T_Y, T_Z)$ but can be regarded as translations in the presence of rotation per unit time.

Using perspective projection and letting f be the focal length, we derive

$$\begin{aligned} x(t) &= \frac{f(ta_X + X - t\Omega_Z Y + t\Omega_Y Z)}{ta_Z - t\Omega_Y X + t\Omega_X Y + Z} = \frac{f(ta_X f/Z + x_0 - t\Omega_Z y_0 + t\Omega_Y f)}{ta_Z f/Z - t\Omega_Y x_0 + t\Omega_X y_0 + f} \\ y(t) &= \frac{f(ta_Y + t\Omega_Z X + Y - t\Omega_X Z)}{ta_Z - t\Omega_Y X + t\Omega_X Y + Z} = \frac{f(ta_Y f/Z + t\Omega_Z x_0 + y_0 - t\Omega_X f)}{ta_Z f/Z - t\Omega_Y x_0 + t\Omega_X y_0 + f} \end{aligned} \quad (4)$$

With the small motion assumption, we derive the image motion equation from the brightness constancy relation, i.e., $I(x(t), y(t), t) = I(x_0, y_0, 0) = F(x_0, y_0)$,

$$I(x, y, t) = F(x + t(\alpha + \gamma x + \rho y + \delta x^2 + \epsilon xy), y + t(\beta - \rho x + \gamma y + \delta xy + \epsilon y^2)) \quad (5)$$

$$\text{where } \alpha = -f \left(\frac{a_X}{Z} + \Omega_Y \right), \beta = -f \left(\frac{a_Y}{Z} - \Omega_X \right), \gamma = \frac{a_Z}{Z}, \rho = \Omega_Z, \delta = -\frac{1}{f} \Omega_Y, \epsilon = \frac{1}{f} \Omega_X. \quad (6)$$

* In an observer-centered coordinate frame; Z is the axis along the line of sight.

From this equation, we see that the local image motion is quadratic, However, the assumptions of small rotation and reasonable focal length f allow us to simplify to a linear model,

$$I(x, y, t) = F(x + t(\alpha + \gamma x + \rho y), y + t(\beta - \rho x + \gamma y)) \quad , \quad (7)$$

which will be used in the following derivation.

3.2 Spatial and temporal image derivatives

We estimate image spatial and temporal derivatives with Hermite polynomial differentiation filters [13] [23] . These filters are orthogonal and their Gaussian derivative properties provide the numerical stability required. We use this filtering scheme and the motion equation (7) to derive a linear system of equations. In linear least square form,

$$E = \min \|As + b\|, \text{ where} \quad (8)$$

$$s = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \rho \end{bmatrix}, b = \begin{bmatrix} \hat{I}_t \\ \hat{I}_{xt} \\ \hat{I}_{yt} \\ \hat{I}_{xxt} \\ \hat{I}_{xyt} \\ \hat{I}_{yyt} \end{bmatrix}, A = \begin{bmatrix} \hat{I}_x & \hat{I}_y & \sigma^2(\hat{I}_{xx} + \hat{I}_{yy}) & 0 \\ \hat{I}_{xx} & \hat{I}_{xy} & \sigma^2(\hat{I}_{xxx} + \hat{I}_{xyy}) + \hat{I}_x & -\hat{I}_y \\ \hat{I}_{xy} & \hat{I}_{yy} & \sigma^2(\hat{I}_{xxy} + \hat{I}_{xxx}) + \hat{I}_y & \hat{I}_x \\ \hat{I}_{xxx} & \hat{I}_{xxy} & \sigma^2(\hat{I}_{xxxx} + \hat{I}_{xxyy}) + 2\hat{I}_{yy} & -2\hat{I}_{xy} \\ \hat{I}_{xxy} & \hat{I}_{xyy} & \sigma^2(\hat{I}_{xxxy} + \hat{I}_{xyyy}) + 2\hat{I}_{xy} & \hat{I}_{xx} - \hat{I}_{yy} \\ \hat{I}_{xyy} & \hat{I}_{yyy} & \sigma^2(\hat{I}_{xxyy} + \hat{I}_{yyyy}) + 2\hat{I}_{yy} & 2\hat{I}_{xy} \end{bmatrix}, \quad (9)$$

where the \hat{I} 's are derivatives computed by Hermite polynomial filters and σ is the standard deviation of the $G(x)$ used in defining the Hermite polynomials. Note that the higher order Hermite filter outputs are relatively small [24] , the above matrix A can be simplified as

$$A = \begin{bmatrix} \hat{I}_x & \hat{I}_y & 0 & 0 \\ \hat{I}_{xx} & \hat{I}_{xy} & \hat{I}_x & -\hat{I}_y \\ \hat{I}_{xy} & \hat{I}_{yy} & \hat{I}_y & \hat{I}_x \\ \hat{I}_{xxx} & \hat{I}_{xxy} & 2\hat{I}_{yy} & -2\hat{I}_{xy} \\ \hat{I}_{xxy} & \hat{I}_{xyy} & 2\hat{I}_{xy} & \hat{I}_{xx} - \hat{I}_{yy} \\ \hat{I}_{xyy} & \hat{I}_{yyy} & 2\hat{I}_{yy} & 2\hat{I}_{xy} \end{bmatrix}. \quad (10)$$

The above simplification is also supported by the fact that the higher order Gaussian derivatives are usually smaller and less accurate. In other words, such an approximation does not induce much error.

It is necessary that we derive the motion constraint equations up to the third order for the purpose of motion boundary extraction because we need to have more constraints than unknowns to obtain a least square formulation and compute the residual, E in (8). The residual measures the amount of disagreement among the equations in a linear system. In other words, if these equations are derived from a mathematical model, then the residual reflects the deviation from the underlying assumptions of the model. In our case, the assumptions are brightness constancy and local coherent motion. In the following subsection, we analyze the relationship between the residual and the motion boundary.

3.3 Analytical relations between the residual and the motion boundary

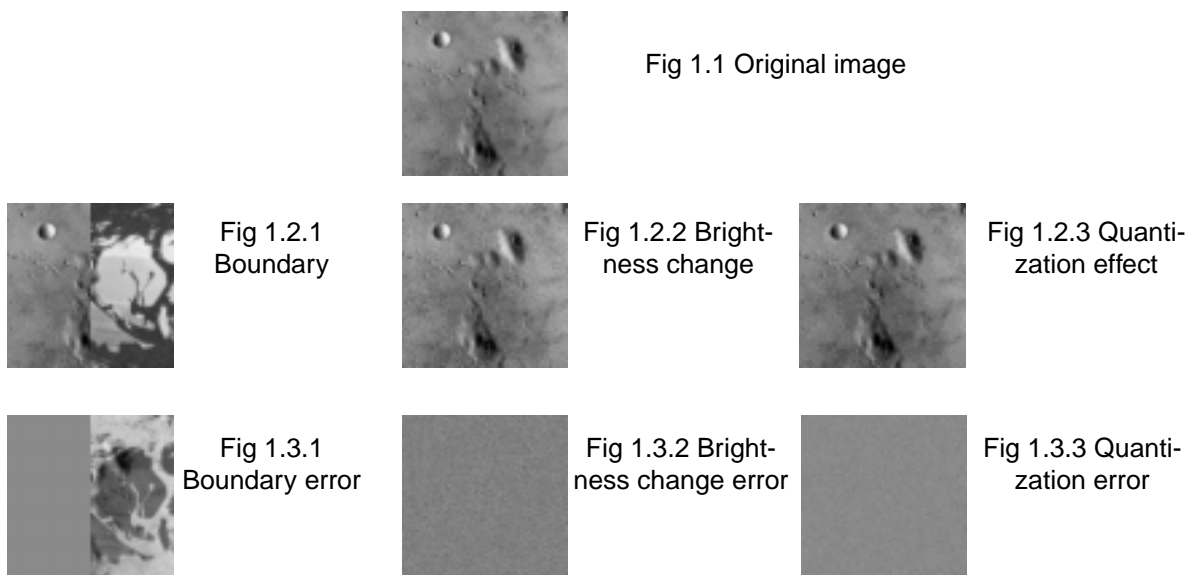
The residual of our algorithm is $E = \min \|As + b\|$. The residual error can result from the approximation errors of our computational model in describing the physical world. Specifically, these errors are:

1. The assumption of the motion model is violated in the local window, i.e., the window covers more than one moving object. Occlusion and multiple independently moving objects in a window can cause this problem.
2. The assumption of constant image brightness is violated, i.e., the image intensity pat-

tern changes over time due to sensor noise, change in the viewing angle, shadows, etc.

3. Quantization or truncation error. Quantization errors result from digitization of the image intensities and sampling of the Hermite polynomial filters. Truncation errors are introduced when we use a limited spatial support to compute $\{\hat{I}\}$. Within the small window, the Hermite polynomials are no longer orthogonal and the derivatives computed are not accurate. This situation is worse for higher order differentiation filters.

The following figures show the typical effects and magnitudes of errors due to motion boundaries, brightness changes, and quantization in a small local neighborhood. Fig 1.1 shows the image at time 0. Fig 1.2.1 shows the same patch occluded by another patch. Fig 1.2.2 shows the patch undergoing random brightness change due to sensor noise. Fig 1.2.23 shows the intensity of the same patch quantized coarsely. Fig 1.3.1-Fig 1.3.3 show the magnitudes of the errors associated with these situations, where mid-gray represents 0, brighter levels mean positive and darker levels mean negative. It can be seen that motion boundaries usually induce the largest errors among the three. Note that the figures only show the errors in the image sequence. After we apply



the differentiation filters, the errors are transformed into errors in the linear system.

Hence we can model the above errors as perturbations or noise to the linear system [22] :

$$\tilde{E} = \min\|(A + N)\tilde{s} + (b + \Delta b)\| \quad , \text{ where } N \text{ and } \Delta b \text{ denote errors.} \quad (11)$$

We derive the analytical relationship between the residual and the errors as follows. Let A and b , defined in (9), contain no noise. Then

$$E = As + b = 0 \text{ and } s = -(A^T A)^{-1} A^T b. \quad (12)$$

Let the noise contaminated optical flow be \tilde{s} and the new residual be \tilde{E} , and assume that $N \ll A$ and $\Delta b \ll b$ elementwise, i.e., $NN^T = 0$ and $N\Delta b = 0$. Then

$$\tilde{s} = -[(A + N)^T(A + N)]^{-1}(A + N)^T(b + \Delta b) \quad , \text{ and} \quad (13)$$

$$\begin{aligned} [(A + N)^T(A + N)]^{-1} &\approx (A^T A [I + (A^T A)^{-1}(A^T N + N^T A)])^{-1} \\ &\approx [I - (A^T A)^{-1}(A^T N + N^T A)](A^T A)^{-1} \quad , \text{ so} \end{aligned} \quad (14)$$

$$\tilde{s} \approx -(A^T A)^{-1} A^T b + (A^T A)^{-1}(A^T N + N^T A)(A^T A)^{-1} A^T b - (A^T A)^{-1} N^T b - (A^T A)^{-1} A^T \Delta b$$

Using (12), this can be simplified as follows:

$$\tilde{s} \approx s - (A^T A)^{-1} A^T N s - (A^T A)^{-1} A^T \Delta b \text{ and } \Delta s \approx -(A^T A)^{-1} A^T N s - (A^T A)^{-1} A^T \Delta b.$$

For the residual, substituting \tilde{s} into (11), and using (12), we derive

$$\begin{aligned} \tilde{E} &\approx \|(A + N)s - A(A^T A)^{-1} A^T N s - A(A^T A)^{-1} A^T \Delta b + b + \Delta b\| \\ &\approx \|(I - A(A^T A)^{-1} A^T)(N s + \Delta b)\| \quad . \end{aligned} \quad (15)$$

Further analysis shows that expression $I - A(A^T A)^{-1} A^T$, denoted by T , has only two non-trivial eigenvalues, both 1. We thus conclude that \tilde{E} is proportional to the noise magnitude and noise orientation with respect to the matrix T . And since T is dependent on the image intensity pattern, which we cannot separate from the noise, we will only use the fact that residual error is

proportional to the noise magnitude. But in order to use the residual to extract boundaries we still need to separate the residual error induced by motion boundaries from that by other sources. Therefore we analyze the residual profile in the spatial domain in the following subsection.

3.4 Residual profile

We now show that the residual profile across a motion boundary follows a specific pattern and is very different from the residual profiles arising from brightness changes or quantization errors. We can then use a spatial filter that matches this profile to extract motion boundaries.

Fig 2.1 shows a motion boundary neighborhood. A dotted square represents a local window used to estimate image derivatives and the residual for the center pixel. By sliding the window across the boundary, we can compute and plot the residual profile. A typical residual profile is shown in Fig 2.2. It has a big plateau centered on the motion boundary. The width of the plateau is

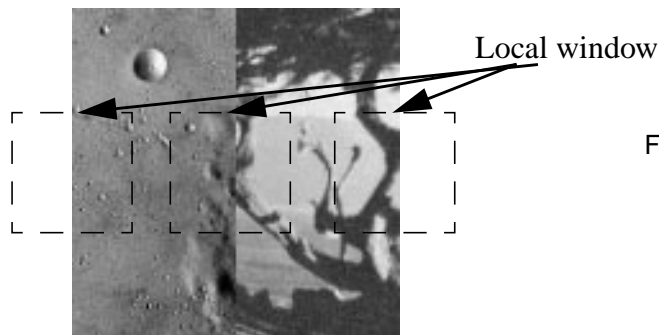


Fig 2.1 Motion boundary

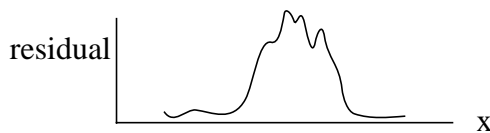


Fig 2.2 Typical residual profile across boundary

about the same as the local window size. This is because only in that region does the local window cover the boundary.

Brightness changes and quantization errors, on the other hand, are usually scattered in the image; we do not expect their residual profiles to look like the residual profiles due to motion boundaries. Also, since residuals arising from motion boundaries are larger than those arising from the other two sources, their profiles should be very prominent.

3.5 Motion boundary extraction based on residual profile

Based on the above findings, we can extract motion boundaries using two spatial filter (for different directions) designed according to Canny’s criteria [6] for wide ridge edge detection. The maxima of the two responses are thresholded to form thick boundaries. On the thick boundaries, we perform a morphological medial axis operation or skeletonization* [11] to extract the center loci of the boundaries. Some simple pruning and contour following are then done to prevent streaking since the medial axis does not guarantee connectivity. Note that we do not use nonmaximum suppression and hysteresis as in [6] . It is because the residual is not proportional to noise† and there may be multiple peaks in the residual profile that will cause the maximum to drift away from the actual center of the ridge. We have verified in experiments that the medial axis provides better localization than nonmaximum suppression and hysteresis. The algorithm is summarized in Fig 3.

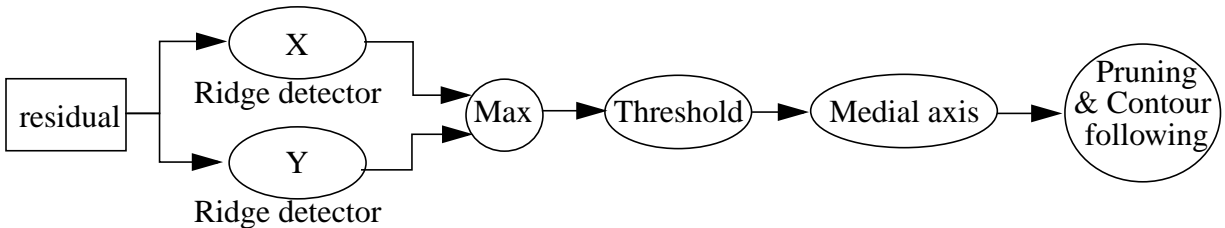


Fig 3. Summary of our boundary detection algorithm.

4. Boundary Properties

One of the nice properties of motion-model-based boundary extraction is that we are able to not only locate boundaries but also find properties of the boundaries using the neighboring model parameters, which are simultaneously estimated. For example, we can find relative depth at the boundary from the neighboring motion parameter $\gamma = a_z/Z$. Under certain circumstances, such relative depth information may be used to perform qualitative scene reconstruction [28] , that is, discriminate the distances of different objects. We can also find the directions of boundaries and

* It is performed using Khoros 1.5 vmskel.

† It depends on the noise vector direction (Section 3.3).

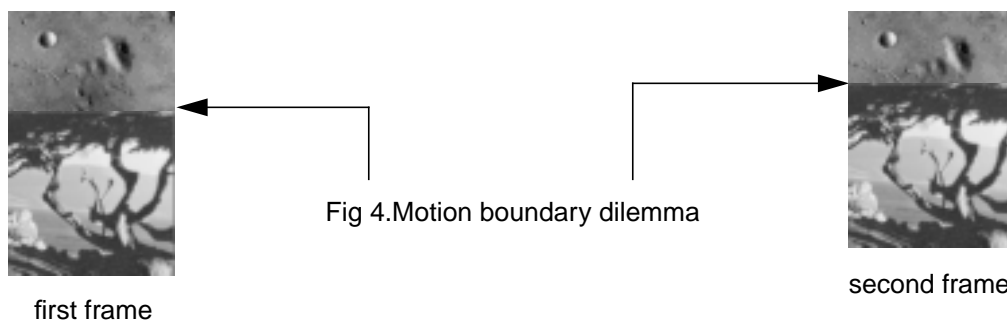
boundary strengths in ways similar to what many edge detection algorithms do. The boundary direction provides valuable information for subsequent processing. For example, it may assist in following boundary contours to avoid streaking, or in updating incorrect optical flow values around the boundary. The boundary strength reveals the relative motion on both sides of the boundary; that is, when the relative motion of the two neighboring objects is large, the response to the boundary detection filter will be large. This property lends itself to various motion applications such as obstacle avoidance or qualitative scene reconstruction.

From the above explanations, it can be seen that these properties will play important roles when we integrate the optical flow algorithm [24] and the boundary extraction algorithm. Note that while these properties are commonly computed by regular edge detectors, motion boundary extraction algorithms rarely deal with them.

5. Evaluation and Experiments

It is very important to evaluate motion boundary extraction separately from optical flow. This makes clear what component of the motion estimation algorithm needs to be improved.

The evaluation of motion boundary extraction is similar to that of edge detection except that the ground truth is less clearly defined in the former case. For example, given only two successive images with motion boundaries (Fig 4.), the motion boundary can be designated at the location



before or after the motion. We therefore use an odd number of frames to extract boundaries. The motion boundary will be defined as the location of the boundary in the center frame.

A good quantitative evaluation scheme for motion boundary extraction should account for the

probabilities of detection and miss as well as the localization error. We reviewed several existing schemes and found that Heyden’s method of evaluation [16] is best suited for motion boundary extraction purposes. It offers the following advantages over Abdou and Pratt’s method[1] : first, it penalizes long streaking, i.e., large gaps of missed boundaries; second, it penalizes thick edges; third, there is no need to perform a search for correspondences between detected and ground truth

motion boundaries. This evaluation scheme is sketched in Fig 5. It involves only binary image

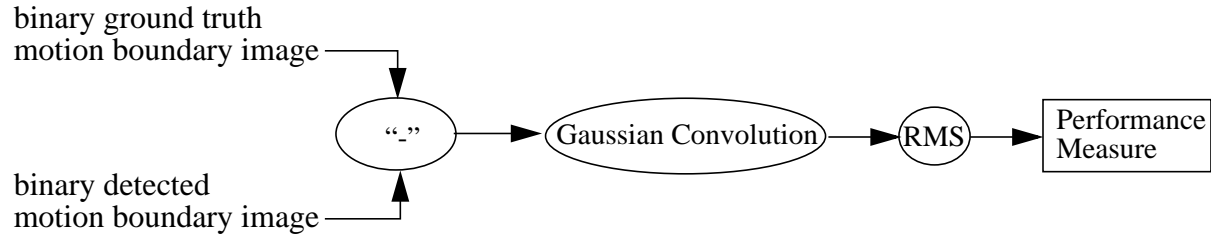


Fig 5. Heyden’s quantitative evaluation scheme.

subtraction, Gaussian convolution, and computing the root mean square (RMS) of all pixels in the resulting image. The performance measure is the RMS of the Gaussian smoothed difference image. The Gaussian convolution is actually the crucial step that achieves the above advantages.

Note that in this scheme, a better algorithm will yield a smaller output quantity, with zero as its minimum.

In order to make comparisons, we also implemented algorithms developed by Schunck [30] and Thompson et al. [33] . In implementing Schunck’s algorithm, we used 3-D Hermite polynomial filters to compute first order derivatives and perform constraint line clustering to estimate optical flow. In fact, we had originally used Gaussian smoothing and Sobel operators to compute the derivatives and found the results too noisy to use. The Canny edge detector is applied to the flow components to find motion boundaries. In implementing Thompson’s algorithm, we used Lucas & Kanade’s algorithm implemented by Barron et al. [3] . The initial flow output is not dense, so we implemented a propagation and smoothing technique to fill the field. After the flow is estimated, we use the vector field discontinuity detector suggested in [33] , using direction reversal of the Laplacian response of the flow vector field, to locate motion boundaries.

The first image we used is shown in Fig 6.1. It is a sequence composed of a baby face travers-



Fig 6.1 Moving face on random dots

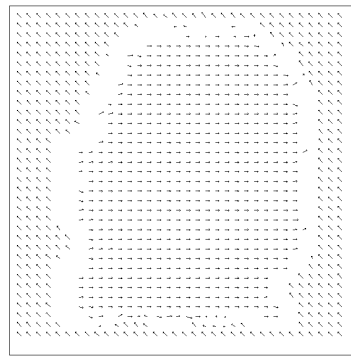


Fig 5.2 Approximate flow field



Fig 5.3 Motion boundary

ing laterally in front of a moving random dot background. The approximate flow map and the motion boundary ground truth are shown in Fig 5.2 and Fig 5.3, respectively. This image sequence is synthesized so as to contain curved motion boundaries, which are common to real world scenes but present difficulties for most motion boundary extraction algorithms. This is because motion boundaries are often wider than intensity edges due to the nature of the motion estimation algorithm and it is very difficult to capture wide as well as high curvature features.

In Fig 6.1-Fig 6.3, we show our algorithm's residual map and Schunck's and Thompson's flow fields. They represent the bases upon which these algorithms extract boundaries. Thomp-



Fig 7.1 Residual map

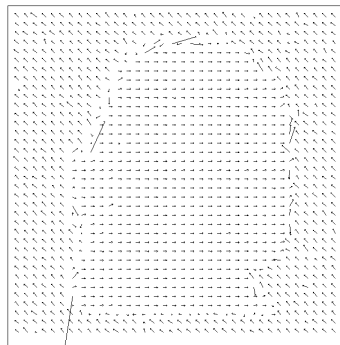


Fig 6.2 Schunck's flow field

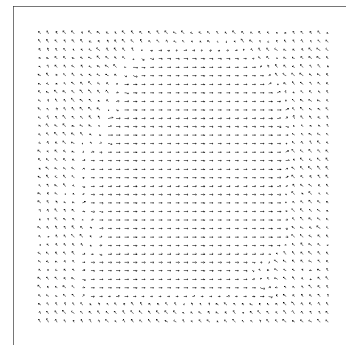


Fig 6.3 Thompson's flow field

son's flow field (Fig 6.3) is smooth across boundaries as expected, while Schunck's flow field (Fig 6.2) is noisier right on boundaries but more accurate near boundaries.

Next we show the detected boundary and true motion boundary for the three algorithms. In

Fig 8.1-Fig 7.3, the dark edge represents the true motion boundary while the white edge represents the boundary detected. Note that when the true boundary is detected, the color of the edge becomes gray. These images are obtained by subtracting the ground truth boundary image from the detected boundary image as dictated by Heyden’s evaluation scheme.



Fig 8.1 Our algorithm’s boundary



Fig 7.2 Schunck’s boundary

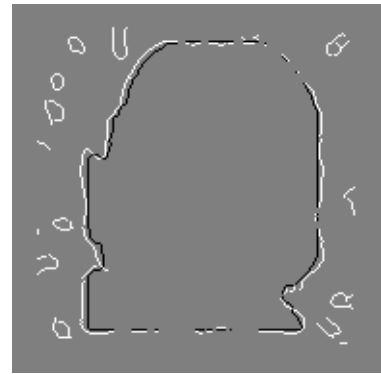


Fig 7.3 Thompson’s boundary

In Fig 7.2, it can be seen that Schunck’s algorithm suffers from boundary drift caused by noise on the boundary as well as localization errors in the corners, as mentioned in [30] . On the other hand, when the motion boundary is a straight line, Schunck’s algorithm performs better than the other two. In Fig 7.3, it can be seen that Thompson’s algorithm suffers from flow noise away from boundaries. Since it uses a direction reversal technique similar to zero crossings, spurious edges are detected. Otherwise, the localization is very good. Our algorithm’s boundary is better at corners and essentially free of the major problems of the other two. The following table summarizes the quantitative performance measure computed by Heyden’s evaluation scheme. It can be seen that our algorithm is better than the other two.

Table 2: Summary of quantitative performance measure

Comparison Algorithms	Our Algorithm	Schunck’s Algorithm	Thompson et al.’s Algorithm
Performance measure	5.85	7.86	10.32

The next image we use is the Yosemite fly-by sequence shown in Fig 9.1. This is a synthesized sequence in which the observer is approaching the scene and motion boundaries exist between

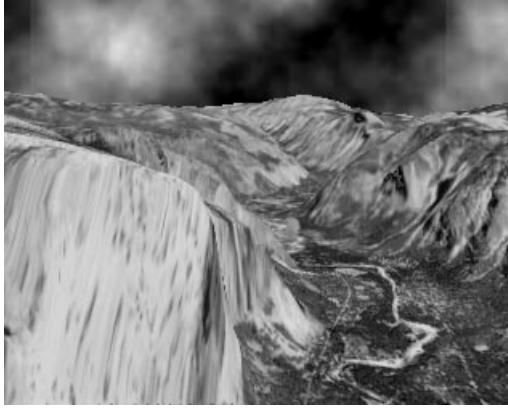


Fig 9.1 Yosemite fly-by

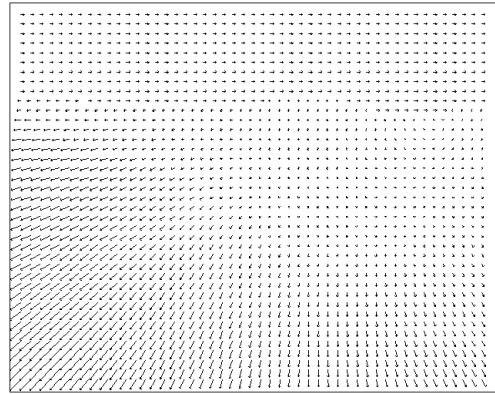


Fig 9.2 Yosemite fly-by flow field

objects of different depths. As can be seen by the flow field (Fig 9.2), two prominent motion boundary curves exist. One separates the sky from the mountains, and the other separates the domed mountain in the lower left corner from the other mountains. The boundary ground truth is not available. In Fig 10.1-Fig 9.3, we show the results of the three boundary extraction algorithms overlaid on the original image. The white edge points represent the extracted boundaries.



Fig 10.1 Our algorithm's result



Fig 9.2 Schunck's result

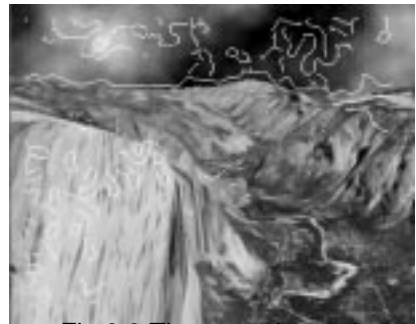


Fig 9.3 Thompson's result

Note that the boundaries that separate sky and mountains are easier to extract because the motion directions are different on the two sides. All three algorithms indeed extract these boundaries. However, the other boundary that separates the domed mountain from the other background mountains is not as easy to extract because the motions on the two sides are in the same direction but have different magnitudes. Note that this kind of motion field is typical in the image sequences captured by a forward moving observer. In Fig 9.2 Schunck's algorithm fails to extract these boundaries because the noise on both sides overwhelms the small variation in flow. In Fig 9.3 Thompson's algorithm fails to extract these boundaries because the presmoothing and filling of

the sparse field smooths out the small flow variation. On the other hand, our algorithm extracts a large part of this boundary curve (Fig 10.1).

Our motion-model-based method offers the capability of segmenting moving objects with different flows, divergences, or curls. The Yosemite fly-by sequence, for example, contains different divergences. The residual values indeed account for incoherence of the above three motion parameters in the local window. This is why our algorithm is capable of extracting these boundaries.

6. Conclusion

Motion boundary extraction algorithms are as important as motion estimation algorithms for the complete motion recovery problem. However, their interdependency poses a computational dilemma that renders any partial solution inaccurate. Indeed, the only way to solve the motion recovery problem is to simultaneously address both motion segmentation and estimation. While recent research has focused on iterative methods, we propose a method based on a general motion model. This method is local, non-iterative, and simultaneously deals with both motion estimation and boundary extraction.

The motion-model-based approach fits the local 3-D image pattern to a motion model and outputs a boundary likelihood measure, the residual, which may be used to extract motion boundaries. Compared with motion boundary extraction from flow, it offers several advantages: first, it is a scalar and thus avoids handling vector field discontinuities; second, the residual is less noisy on the boundary than the flow; third, the residual corresponds to true 3-D motion discontinuities instead of high variations caused by flow field nonuniformity within an object. In fact, the residual accounts for discontinuities in flow, divergence, and curl.

The evaluation of motion boundary extraction should be separated from the evaluation of optical flow to truly understand the performance of the individual motion algorithm components. We employed a simple but elegant evaluation scheme and synthesized a difficult motion sequence for comparison. We then demonstrated that our algorithm performs better than existing algorithms.

In the future, we will integrate the motion estimation algorithm described in [23] and the boundary extraction algorithm developed here. This will involve not only updating the flow around the boundaries but also finding important motion information from boundary properties such as the local relative depth. Other future topics include finding occluding surfaces, and global topological sorting of moving objects. These are important for obstacle avoidance and navigation.

Acknowledgment

The authors would like to thank Dr. Bill Thompson for his advice on implementing his boundary extraction algorithm. We also thank Mr. David Jiang for implementing Schunck's constraint line clustering algorithm.

References

- [1] Abdou, I.E. and Pratt, W.K., "Quantitative Design and Evaluation of Enhancement Thresholding Edge Detectors", *Proceedings of IEEE*, vol. 67, no. 5, 1979.
- [2] Adiv, G., "Inherent Ambiguities in Recovering 3-D Motion and Structure from a Noisy Flow Field", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 477-489, 1989.
- [3] Barron, J. L., Fleet, D. J. and Beauchemin, S. S., "Performance of Optical Flow Techniques", *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43-77, 1994.
- [4] Bergen, J., Burt, P., Hingorani, R. and Peleg, S., "A Three-Frame Algorithm for Estimating Two-Component Image Motion", *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 14, no.9, pp. 886-896, 1992.
- [5] Braddick, O.J., "A Short-Range Process in Apparent Motion", *Vision Research*, vol. 14, pp. 519-527, 1974.
- [6] Canny, J., "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 8, no.11, pp. 679-698, 1986.
- [7] Darrel, T. and Pentland, A., "Robust Estimation of a Multi-Layered Motion Representation", *Proceedings of IEEE Workshop on Visual Motion*, Princeton, NJ, pp. 173-178, 1991.
- [8] Davis, L., Wu, Z. and Sun, H., "Contour-Based Motion Estimation", *University of Maryland TR-*

- 1179, June, 1982.
- [9] Dengler, J., "Estimation of Discontinuous Displacement Vector Fields with the Minimum Description Length Criterion", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lahaina, HI, pp. 276-282, 1991.
 - [10] Fleet, D.J. and Jepson, A.L., "Computation of Component Image Velocity from Local Phase Information", International Journal of Computer Vision, vol. 5, no.1, pp. 77-104, 1990.
 - [11] Giardina, C. and Dougherty, E.R., "Morphological Methods in Image and Signal Processing", Prentice Hall, Englewood Cliffs, NJ, 1988.
 - [12] Hartley, R., "Segmentation of Optical Flow Fields by Pyramid Linking", Pattern Recognition Letters, vol.3, no. 5, pp.253-262, 1985.
 - [13] Hashimoto M. and Sklansky, J. "Multiple-Order Derivatives for Detecting Local Image Characteristics", Computer Vision, Graphics, and Image Processing, vol. 39, pp. 28-55, 1987.
 - [14] Heeger, D. J., "Optical Flow Using Spatiotemporal Filters", International Journal of Computer Vision, vol. 1, no. 4, pp. 279-302, 1988.
 - [15] Heitz, F. and Bouthemy, P., "Multimodal Estimation of Discontinuous Optical Flow Using Markov Ransom Fields", IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 15, no.12, pp. 1217-1232, 1993.
 - [16] Heyden, F., "Evaluation of Edge Detection Algorithms", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.618-622, 1992.
 - [17] Hildreth, E., "The Measurement of Visual Motion", MIT Press, Cambridge, MA, 1984.
 - [18] Horn, B. K. P. and Schunck, B. G., "Determining Optical Flow", Artificial Intelligence, vol. 17, pp. 185-204, 1981.
 - [19] Jepson, A., Black, M., "Mixture Model for Optical Flow Computation", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, pp.760-761, 1993.
 - [20] Kearney, J. K., Thompson, W. B. and Boley, D. L., "Optical Flow Estimation: An Error Analysis of Gradient Based Methods With Local Optimization", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, pp. 229-244, 1987.
 - [21] Koch, C., Marroquin, J. and Yuille, A., "Analog 'Neural' Networks in Early Vision", Proceedings of the National Academy of Sciences, vol. 83, pp. 4263-4267, 1986.

- [22] Liu, H., Hong, T., Herman, M. and Chellappa, R., "A Reliable Optical Flow Algorithm Using 3-D Hermite Polynomials", NIST IR-5333, December, 1993.
- [23] Liu, H., Hong, T., Herman, M. and Chellappa, R., "A Generalized Motion Model for Estimating Optical Flow Using 3-D Hermite Polynomials", Proceedings of the International Conference on Pattern Recognition, Jerusalem, Israel, pp. 360-366, 1994.
- [24] Liu, H., Hong, T., Herman, M. and Chellappa, R., "A General Motion Model and Spatio-temporal Filters for Computing Optical Flow", University of Maryland-TR -3365, November 1994.
- [25] Murray, D. and Buxton, B., "Scene Segmentation from Visual Motion Using Global Optimization", IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 9, no.2, pp. 220-228, 1987.
- [26] Mutch, K. and Thompson, W., "Analysis of Accretion and Deletion at Boundaries in Dynamic Scenes", IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 7, no.2, pp. 133-138, 1985.
- [27] Nakayama, K. and Loomis, J.M., "Optical Velocity Patterns, Velocity Sensitive Neurons, and Space Perception: a Hypothesis", Perception, vol. 3, 1974.
- [28] Nelson, R. and Aloimonos, Y., "Obstacle Avoidance Using Flow Field Divergence", IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 11, no.10, pp. 1102-1106, 1989.
- [29] Potter, J. L., "Velocity as a Cue to Segmentation Using Motion Information", IEEE Transactions on Systems, Man, Cybernetics, vol. 5, pp. 390-395, 1975.
- [30] Schunck, B., "Image Flow Segmentation and Estimation by Constraint Line Clustering", IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 11, no.10, pp. 1010-1027, 1989.
- [31] Shizawa, M. and Mase, K., "A Unified Computational Theory for Motion Transparency and Motion Boundaries Based on Eigenenergy Analysis", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lahaina, HI, pp. 289-295, 1991.
- [32] Spoerri, A. and Ullman, S., "The Early Detection of Motion Boundaries", A.I. Memo no. 935, AI Lab, MIT, 1987.
- [33] Thompson, W., Mutch, K. and Berzins, V., "Dynamic Occlusion Analysis in Optical Flow Fields", IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 7, no.4, pp. 374-383, 1985.
- [34] Verri, A. and Poggio, T. "Motion Field and Optical Flow: Qualitative Properties", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 5, pp. 490-498, 1989.

- [35] Werkhoven, P. and Koenderink, J.J., "Extraction of Motion Parallax Structure in the Visual System I", *Biological Cybernetics*, vol. 63, pp. 185-191, 1990.