COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 70, No. 1, April, pp. 87–100, 1998
ARTICLE NO. IV970625

Motion-Model-Based Boundary Extraction and a Real-Time Implementation*

Hongche Liu†

Intelligent Systems Division, National Institute of Standards and Technology (NIST), Building 220, Room B124, Gaithersburg, Maryland 20899; and Center for Automation Research/Department of Electrical Engineering, University of Maryland, College Park, Maryland 20742

Tsai-Hong Hong‡ and Martin Herman§

Intelligent Systems Division, National Institute of Standards and Technology (NIST), Building 220, Room B124, Gaithersburg, Maryland 20899

and

Rama Chellappa||

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

Received February 13, 1996; accepted March 17, 1997

Motion boundary extraction and optical flow computation are two subproblems of the motion recovery problem that cannot be solved independently of one another. These two problems have been treated separately. A popular recent approach uses an iterative scheme that consists of motion boundary extraction and optical flow computation components and refines each result through iteration. We present a local, noniterative algorithm that simultaneously extracts motion boundaries and computes optical flow. This is achieved by modeling 3-D Hermite polynomial decompositions of image sequences representing the perspective projection of 3-D general motion. Local model parameters are used to determine whether motion should be estimated or motion boundaries should be extracted at the neighborhood. A definite advantage of this noniterative algorithm is its efficiency. It is demonstrated by a real-time implementation and supporting experimental results. © 1998 Academic Press

Key Words: **motion analysis; segmentation; real-time implementation.**

1. INTRODUCTION

This paper studies the strengths and weaknesses of recent motion boundary detection and motion segmentation algorithms and proposes a local, noniterative algorithm for motion boundary detection as well as its 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. The flow portion of the algorithm is presented in another paper [28].

In general, the *motion recovery* problem involves two major subproblems: *optical flow computation* and *motion segmentation*. Optical flow computation quantitatively measures the motion associated with perceived objects; motion segmentation, on the other hand, qualitatively distinguishes different moving objects. The fact that they are dependent on one another, as described in the following paragraph, has complicated the general motion recovery problem.

Due to the aperture problem, early motion estimation algorithms [21, 22] usually enforced a smooth flow field as an additional constraint. Recent approaches use spatio-temporal filters [13, 18, 28], 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 [45].) Due to the aforementioned optical flow error around motion boundaries, the requirement of a dense flow field, and noise in the optical flow field, motion boundaries are very difficult to extract and/or locate from optical flow. Researchers have used other image cues, for example, accretion and deletion [34], or normal flow [21], to detect motion discontinuities, but 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.

* The U.S. Government's right to retain a nonexclusive royalty-free license in and to the copyright covering this paper, for governmental purposes, is acknowledged.

† E-mail: liu@cme.nist.gov.

‡ E-mail: hongt@cme.nist.gov.

§ E-mail: herman@cme.nist.gov.

|| E-mail: chella@eng.umd.edu.

TABLE 1
Similarities between Model-Based Boundary Extraction and Edge Detection

Method	Motion-model-based boundary extraction	Edge detection
Image intensity model	Translation, expansion, and rotation	Step edge, ramp edge, or ridge edge
Filtering	3-D Hermite polynomial filters	Directional Gaussian derivatives
Model parameters used	Residual	S/N ratio
Postprocessing	Residual profile matching	Nonmaximum suppression, hysteresis

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 take the approach to producing both outputs at the same time.

Our idea is to understand and analyze the image intensity pattern in a sequence generated by a perspective camera undergoing 3-D motion relative to the objects in the scene. The objects themselves may be moving. The resulting nonlinear motion equation is quite complex. Fortunately, we discovered [29] that Hermite polynomial decomposition of the images enables us to translate the complex nonlinear motion equation to a set of linear motion constraint equations of image decompositions of different orders. Conceptually, the Gaussian derivative behavior of Hermite polynomial filters mimics the visual receptive fields [15, 47] of primates. Algorithmically, the separability of Hermite polynomial filters facilitates an efficient real-time implementation. Using a least-square error method on the overdetermined linear system of motion constraint equations, we compute the motion parameters and the residual. An analytical study 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, in theory, it corresponds to real motion boundaries and not flow variation due to rotation or expansion.

In fact, our concept of motion boundary extraction in a spatio-temporal image is similar to edge detection [8] in a spatial image. Both methods are based on a model for the image intensity and noise. Both use image filtering techniques initially to extract model parameters and then use the parameters to locate the likely candidates for discontinuities. Both perform some post-processing to refine the results. The similarities are summarized in Table1 and illustrated in Fig. 1. Motion boundary extraction is more difficult than edge detection because the motion model needs to include a combination of translation, expansion, *and* rotation, while the edge detector needs only a single image discontinuity model such as step edge, ramp edge, *or* ridge edge.

The appeal of a local, noniterative 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 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 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. It was originally developed for edge detection [20].

Our algorithm is designed with a real-time implementation in mind. With the 3-D Hermite polynomial separable filters and a recursive ridge filter design inspired by Sarkar and Boyer [39], as will be explained in Section 5, we have implemented an algorithm on a general purpose machine (80 MHz HyperSparc 10 Themis board) using a DataCube MV200 image processing board for digitizing and subsampling images. The running speed is about 4–5 frames per second on 64 × 64 images. In addition to motion boundary extraction, the optical flow is also computed.

Our work is indebted to many previous studies referred to in Fig. 2. An arrow in the diagram represents an idea (as annotated) extracted from the work.

The rest of the paper is organized as follows. Section 2 surveys previous methods and their strengths and weaknesses. The motion model and our algorithm are introduced in Section 3. The evaluation scheme and experimental results based on it

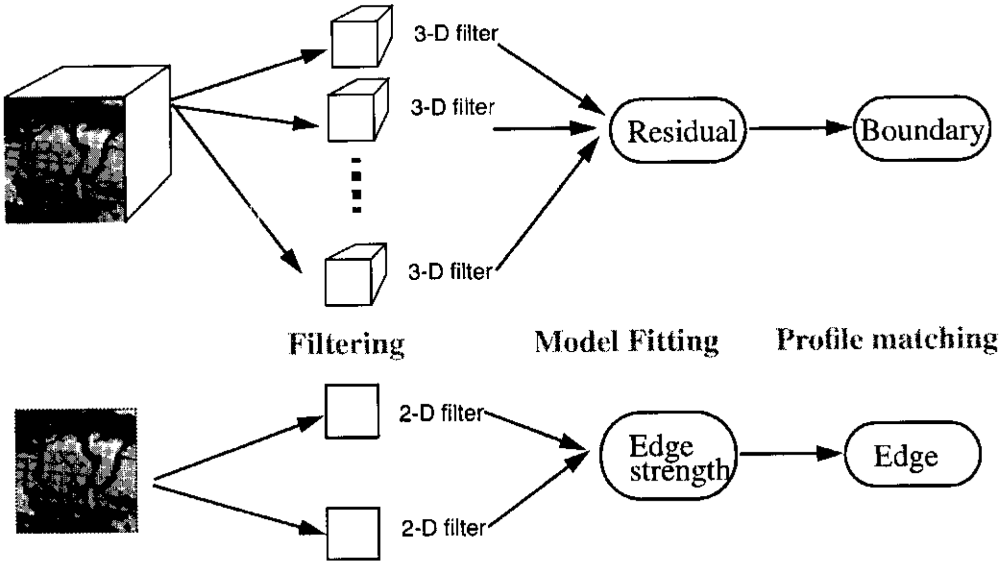

FIG. 1. Analogy between motion-model-based boundary extraction and edge detection.

are presented in Section 4. Section 5 describes the real-time implementation and its results. Section 6 concludes the paper with a statement of our contributions and its potential applications.

2. PREVIOUS WORK

Braddick's psychological experiments on random dot motion [7] set the stage for vision research on motion boundaries. It verified the humans' capability in perceiving boundaries clearly using only motion cues. The existing work on motion boundary extraction or segmentation is summarized in Table 2. This survey is not exhaustive but represents typical work in this area, which will be elaborated in the following subsections.

2.1. Noniterative Algorithms

Early research on motion boundary extraction or motion segmentation can be roughly characterized as based on a noniterative approach. These algorithms can be put into three categories [12, 42] based on whether the motion boundary extraction is performed prior to, simultaneously with, or after the flow field estimation (Refer to Table 2.)

The approaches that extract motion boundaries prior to flow field estimation employ "motion primitives" [42], usually normal flow [21], as a basis. Hildreth's work [21] is one of the first major contributions in the field. The method 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

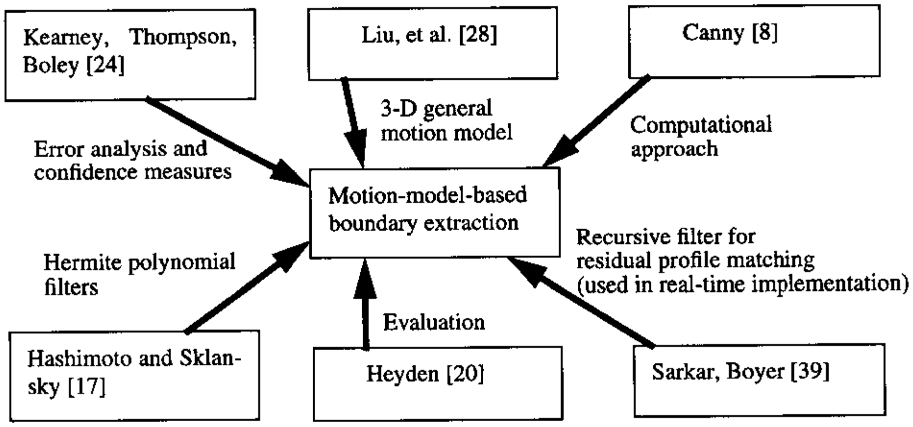
Kearney, Thompson, Boley [24]
Liu, et al. [28]
Canny [8]
3-D general motion model
Error analysis and confidence measures
Computational approach
Motion-model-based boundary extraction
Hermite polynomial filters
Recursive filter for residual profile matching (used in real-time implementation)
Evaluation
Hashimoto and Sklansky [17]
Heyden [20]
Sarkar, Boyer [39]

FIG. 2. Summary of previous work contributing to our work.

TABLE 2
Summary of Current Motion Boundary Extraction Algorithms

Noniterative schemes		Iterative schemes	
Motion boundary extraction vs flow estimation	Algorithm by	Techniques	Algorithm by
Prior to	Hildreth [21], Spoerri and Ullman [42]	Pyramid linking/ Region growing	Hartley [16], Debrunner and Ahuja [11]
Simultaneous with	Mutch and Thompson [34], Schunck [40], Shizawa and Mase [41], Bouthemy [6]	Markov random field with binary line processes	Koch, Marroquin and Yuille [25], Murray and Buxton [32], Heitz and Bouthemy [19], Konrad and Dubois [26]
After	Potter [38], Nakayama and Loomis [36], Adiv [2], Thompson, Mutch, and Berzins [44], Murray and Williams [33], Dengler [12]	Robust estimation	Darrel and Pentland [9], Jepson and Black [23]
		Tracking and nulling	Bergen *et al.* [4]

as the sign of the normal flow changes. The method has two limitations [21]: 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 restrictive for general applications. Spoerri and Ullman [42] pioneered the approach of statistical tests on motion primitives to inferring motion boundaries. 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 [34], [40], and [41]. Mutch and Thompson [34] use the simple and elegant 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. It does not detect, for instance, the boundaries between two neighboring objects moving in parallel directions or a rotating object where no accretion or deletion occurs. Schunck's algorithm [40] makes an important contribution by offering constraint line clustering to break the motion boundary/optical flow dilemma. The basic idea of the algorithm is to use local consensus, instead of smoothing, to determine flow values. 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 algorithm has a tendency to produce high localization error on motion boundary "corners." The clustering technique is heuristic and is prone to numerical instability. Shizawa and Mase [41] are the first to present "multiple-flow constraint equations," a generalization of the common optical flow constraint equation, to deal with motion boundaries and/or transparent motion. Their 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 a 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. Bouthemy's [6] work on moving edge detection extends our understanding of edge detection to the spatio-temporal image domain. To extract motion boundaries from moving edges, one must apply some motion modeling.

Approaches that extract motion boundaries after flow field estimation are more popular. Global techniques such as the Hough transform (Adiv [2]), region growing (Potter [38]), and pyramid linking (Dengler [12]) have been used. Local techniques include center-surround filters (Nakayama and Loomis [36]), direction reversals of the Laplacian operator on the flow vector field (Thompson, *et al.* [44]), and condition number of the linear least-square system by applying the structure from motion model on flow field (Murray and Williams [33]). These approaches offer only a partial solution to the motion estimation problem because the boundary depends heavily on the accuracy of the optical flow. However, they provide algorithms 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 one another 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 noniterative methods but are time-consuming. Note that there are algorithms that use iterative schemes to compute optical flow only. We do not label them as iterative methods here since they do not include the segmentation component.

Hartley's algorithm [16] 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. Debrunner and Ahuja use a region growing technique to segment feature points based on their motion in a long image sequence. If the features are sufficiently dense, this method should be adequate for boundary extraction.

The use of a Markov random field model for flow has been proposed by Koch *et al.* [25], Murray and Buxton [32], Heitz and Bouthemy [19], and Konrad and Dubois [26]. They handle flow discontinuities by introducing a binary line process to discourage smoothing across boundaries. 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 and Pentland [9], Jepson and Black [23], and Black and Anandan [5]. They use a multilayered motion model ("mixture model" [23]) 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 "nulling" mechanism is used to separate and estimate individual motions. In other words, 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 and they usually solve the global motion segmentation problem, 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 3-D motion model. The necessary elements of the scheme are a motion model which is based on arbitrary 3-D motion, a method for estimating image properties, for which we use Hermite polynomial decomposition of images, and a procedure to extract motion model irregularities as boundaries. The following subsections briefly present the derivations of these three elements; the details can be found in [29] and [28].

3.1. The Computational Model

We have developed a general motion model by considering any point $\vec{P} = (X, Y, Z)^T$ in a 3-D space undergoing steady rotation $(\Omega_X, \Omega_Y, \Omega_Z)$ and translation (T_X, T_Y, T_Z) per unit time as seen from a viewer by perspective projection with focal length f. We derive [28]

$$\begin{aligned} I(x, y, t) = F(x + t(\alpha + \gamma x + \rho y + \delta x^2 + \varepsilon xy),\\ y + t(\beta - \rho x + \gamma y + \delta xy + \varepsilon y^2)), \end{aligned} \tag{1}$$

where

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

This quadratic image motion equation can be simplified to the following affine model [35, 46] with the assumption of small rotation and/or reasonable focal length

$$I(x, y, t) = F(x + t(\alpha + \gamma x + \rho y), y + t(\beta - \rho x + \gamma y)). \tag{2}$$

We then use Hermite polynomial decompositions of the image to represent the local image properties.

The nth Hermite polynomial $H_n(x)$ is a solution of

$$\frac{d^2 H_n}{dx^2} - 2x\frac{dH_n}{dx} + 2nH_n = 0. \tag{3}$$

The $H_n(x)$ are derived by Rodrigues' formula [17]

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}. \tag{4}$$

By substituting $G(x)$ (with variance σ^2) for e^{-x^2} in (4), we generalize to Hermite polynomials with respect to the Gaussian function. Let these Hermite polynomials be denoted by $\bar{H}_n(x)$.

The scalar product of two functions and the L_2-norm of a function with $G(x)$ as a weight function are defined as

$$\langle a, b\rangle \equiv \int_{-\infty}^{\infty} G(x)a(x)b(x)\,dx \quad \text{and} \quad \|a\| \equiv \langle a, a\rangle^{1/2}. \tag{5}$$

The orthogonality of $\{\bar{H}_n(x)\}$ can be expressed [17]

$$\langle \bar{H}_m, \bar{H}_n\rangle = \sigma^{-2n} n!\delta_{mn}. \tag{6}$$

The 3-D case of Hermite polynomials is especially simple because they are separable

$$\bar{H}_{ijk}(x, y, t) = \bar{H}_i(x)\cdot \bar{H}_j(y)\cdot \bar{H}_k(t). \tag{7}$$

We use the following Gaussian derivative theorem to derive motion constraint equations.

Theorem 1. *A one-dimensional signal $I(x)$ can be decomposed in terms of Hermite polynomials as*

$$I(x) = \sum_{k=0}^{\infty} I_k \frac{\bar{H}_k(x)}{\|\bar{H}_k\|^2}. \tag{8}$$

Then

$$I_k = \langle I, \bar{H}_k\rangle = \langle I^{(k)}, \bar{H}_0\rangle,$$

where

$$I^{(k)} = \frac{d^k I}{dx^k}.$$

Expand both sides of Eq. (2) with Hermite polynomials,

$$\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{k=0}^{\infty} I_{ijk}\frac{\bar{H}_{ijk}}{\|\bar{H}_{ijk}\|^2} = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{k=0}^{\infty} F_{ijk}\frac{\bar{H}_{ijk}}{\|\bar{H}_{ijk}\|^2}$$

then

$$I_{ijk} = \langle I, \bar{H}_{ijk}\rangle = F_{ijk} = \langle F, \bar{H}_{ijk}\rangle \tag{9}$$

due to orthogonality.

Equating I_{ij1} to F_{ij1} and using Theorem 1, we derive (see [29] for details)

$$\begin{aligned} I_{ij1} \approx{}& \alpha I_{(i+1)j0} + \beta I_{i(j+1)0} + \gamma(i + j)I_{ij0} \\ &+ \rho(iI_{(i+1)(j-1)0} + jI_{(i-1)(j+1)0}) \\ &+ \gamma\sigma^2(I_{(i+2)j0} + I_{i(j+2)0}) \\ &+ \rho\sigma^2 I_{(i+1)(j+1)0}. \end{aligned} \tag{10}$$

It is clear from Theorem 1 that the Hermite polynomial decompositions are computed by filtering the image with Hermite polynomial filters. Within a 3-D local window, we estimate $\{I_{ijk}\}$ with the discrete approximation $\{\hat{I}_{ijk}(x, y, t)\}$, that is, the 3-D convolution of the sampled Hermite polynomial filters with the image sequence. We use this filtering scheme and the motion equation (10) to derive a linear system of equations. In linear least squares form,

$$E = \min\|As + b\|, \tag{11}$$

where

$$s = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \rho \end{bmatrix}, \quad b = \begin{bmatrix} \hat{I}_{001} \\ \hat{I}_{101} \\ \hat{I}_{011} \\ \hat{I}_{201} \\ \hat{I}_{111} \\ \hat{I}_{021} \end{bmatrix},$$

$$A = \begin{bmatrix} \hat{I}_{100} & \hat{I}_{010} & \sigma^2(\hat{I}_{200} + \hat{I}_{020}) & 0 \\ \hat{I}_{200} & \hat{I}_{110} & \sigma^2(\hat{I}_{300} + \hat{I}_{120}) + \hat{I}_{100} & -\hat{I}_{010} \\ \hat{I}_{110} & \hat{I}_{101} & \sigma^2(\hat{I}_{210} + \hat{I}_{300}) + \hat{I}_{010} & \hat{I}_{100} \\ \hat{I}_{300} & \hat{I}_{210} & \sigma^2(\hat{I}_{400} + \hat{I}_{220}) + 2\hat{I}_{200} & -2\hat{I}_{110} \\ \hat{I}_{210} & \hat{I}_{120} & \sigma^2(\hat{I}_{220} + \hat{I}_{130}) + 2\hat{I}_{11} & \hat{I}_{200} - \hat{I}_{020} \\ \hat{I}_{120} & \hat{I}_{030} & \sigma^2(\hat{I}_{310} + \hat{I}_{040}) + 2\hat{I}_{020} & 2\hat{I}_{110} \end{bmatrix}, \tag{12}$$

The higher order decompositions are not used because they are relatively small [28] and susceptible to noise in the image.

The residual of our algorithm is $E = \min\|As + b\|$. The residual error results from the approximation errors of our computational model in describing the physical world. Specifically, the causes of 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 pattern 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 $\{\hat{I}\}$.

We model the above errors as perturbations or noise to the linear system [27]

$$\tilde{E} = \min\|(A + N)\tilde{s} + (b + \Delta b)\|, \tag{13}$$

where N and Δb denote errors.

We prove in Appendix A that the magnitude of the residual $\tilde{E}$ is linearly proportional to the noise magnitude $\|Ns + \Delta b\|$

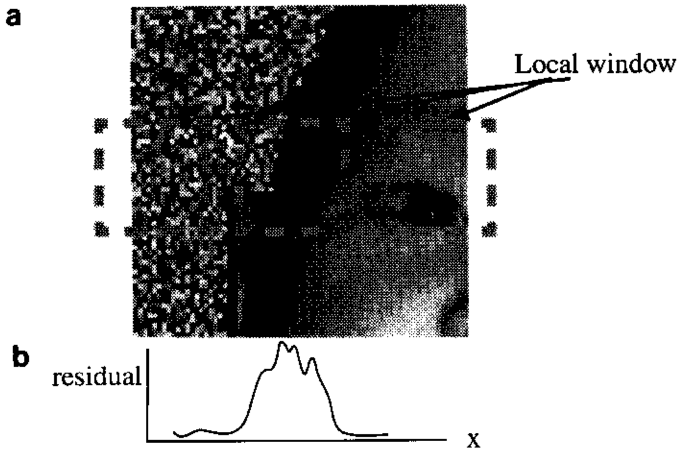

FIG. 3. (a) Motion boundary, (b) typical residual profile across boundary.

according to

$$\tilde{E} \approx \|(I - A(A^TA)^{-1}A^T)\|\|Ns + \Delta b\|. \tag{14}$$

Now that we have established the relation between residual and noise, we show how to use the concept of the residual profile to relate noise with motion boundaries in this model-based approach.

3.2. Residual Profile

We 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 then use a spatial filter that matches this profile to extract motion boundaries.

Figure 3a shows a motion boundary neighborhood. A dashed square represents a local window used to compute Hermite polynomial decompositions 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. 3b. It has a big plateau centered on the motion boundary. The width of the plateau is 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; it is much less probable for them to form residual profiles like those of 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.

TABLE 3
Summary of Quantitative Performance Measure

Algorithms	Our algorithm	Schunck's algorithm	Thompson *et al.*'s algorithm
Performance measure	5.85	7.86	10.32

Based on the above findings, we can extract motion boundaries using a ridge edge detector [8].

4. EVALUATION AND EXPERIMENTS

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

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 [20] is best suited for motion boundary extraction purposes. It has the following nice properties. 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. 4. The performance measure is the RMS of the Gaussian smoothed difference image. The Gaussian convolution of the difference image is 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 [40] and Thompson *et al.* [44]. The Hermite polynomial filter size used in our method is 21 × 21 × 7 (in x, y, t).

The first image we used is shown in Fig. 5a. It is a sequence composed of a baby face traversing laterally in front of a moving random dot background. The approximate flow map and

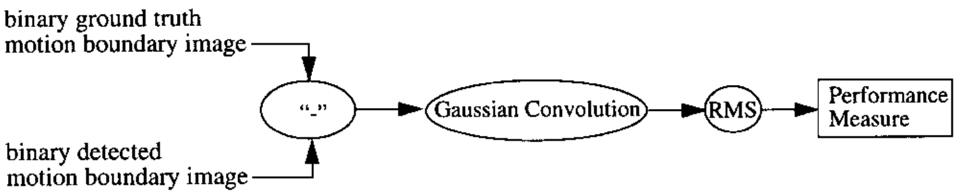

FIG. 4. Heyden's quantitative evaluation scheme.

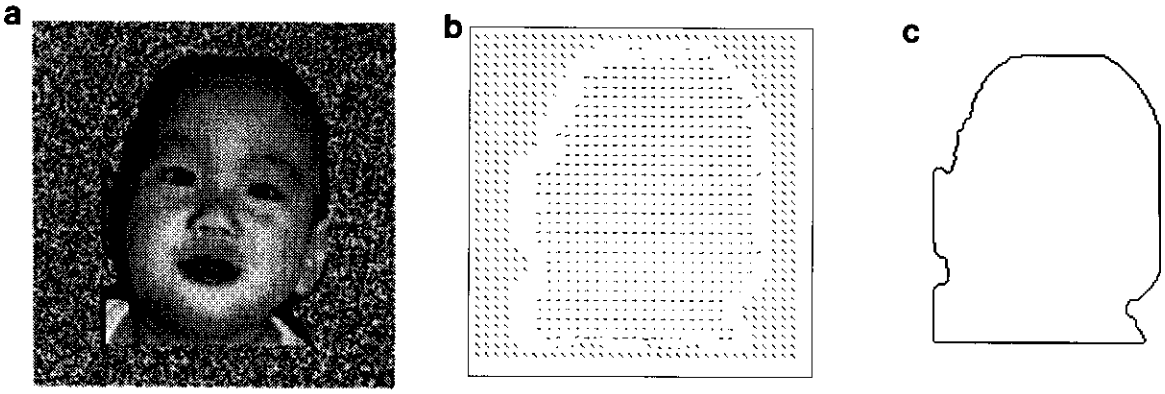

FIG. 5. (a) Moving face on random dots, (b) approximate flow field, (c) motion boundary.

the motion boundary ground truth are shown in Figs. 5b and 5c, respectively. This image sequence is synthesized to contain curved motion boundaries, which are common to real world scenes but present difficulties for most motion boundary extraction algorithms.

In Fig. 6, 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. Thompson's flow field (Fig. 6c) is smooth across boundaries as expected, while Schunck's flow field (Fig. 6b) 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. 7, 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.

In Fig. 7b, 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 [40]. On the other hand, when the motion boundary is a straight line, Schunck's algorithm performs better than the other two. In Fig. 7c, 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. Table 3 summarizes the quantitative performance measure computed by Heyden's evaluation scheme.

The next image we use is the Yosemite fly-by sequence shown in Fig. 8. This is a synthesized sequence in which the observer is approaching the scene and motion boundaries exist between objects at different depths. As can be seen by the flow field (Fig. 8b), 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. 9, we show the results of the three boundary extraction algorithms overlaid on the original image. The white edge points represent the extracted boundaries.

All three algorithms indeed extract the boundaries that separate sky and mountains. Our algorithm is no better than the other

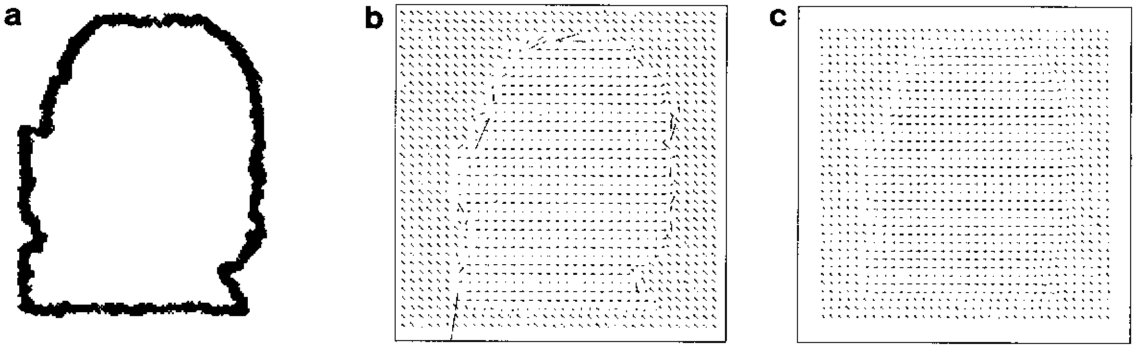

FIG. 6. (a) Residual map, (b) Schunck's flow field, (c) Thompson's flow field.

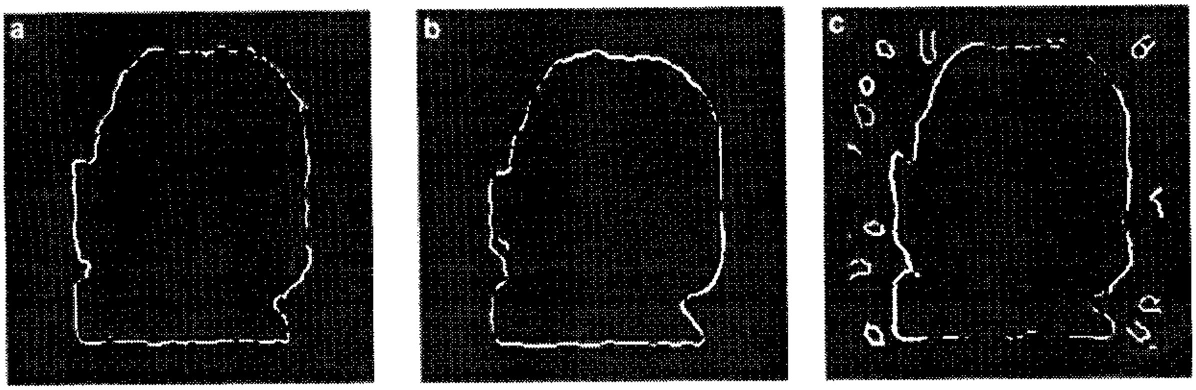

FIG. 7. (a) Our algorithm's boundary, (b) Schunck's boundary, (c) Thompson's boundary.

two in this area. However, the other boundary that separates the domed mountain from the 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. 9b Schunck's algorithm fails to extract these boundaries because the noise on both sides overwhelms the small variation in flow. In Fig. 9c 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 some portion of this boundary curve (Fig. 9a).

5. REAL-TIME IMPLEMENTATION

The Hermite polynomial decomposition is efficient due to its separability (6). We have exploited this property to design a real-time algorithm. Its implementation is illustrated in Fig. 10. The Datacube MV200 is employed because it can digitize and subsample the image at frame rate. All other processing is run on an 80 MHz HyperSparc 10 board. The design of a recursive ridge edge detector is the other aspect of the algorithm that makes the implementation efficient.

The recursive ridge edge detector is based on the function,

$$\begin{aligned} f(t) &= (1 + bt - b^2t^2)e^{-bt} && \text{for } t > 0 \\ \text{and } f(t) &= f(-t) && \text{for } t \leq 0. \end{aligned} \tag{15}$$

As illustrated in Fig. 11, we design this function to have a slope of 0 at the origin and a zero-crossing at $(\sqrt{5} + 1)/2b$, which can be adjusted by changing b. It emulates the optimal symmetric ridge edge detector by Canny [8].

It is a recursive filter because we can derive a recursive relation between input $x(t)$ and output $y(t)$. Considering only the causal component of the filter function, i.e., $f(t)$ for $t > 0$, using the

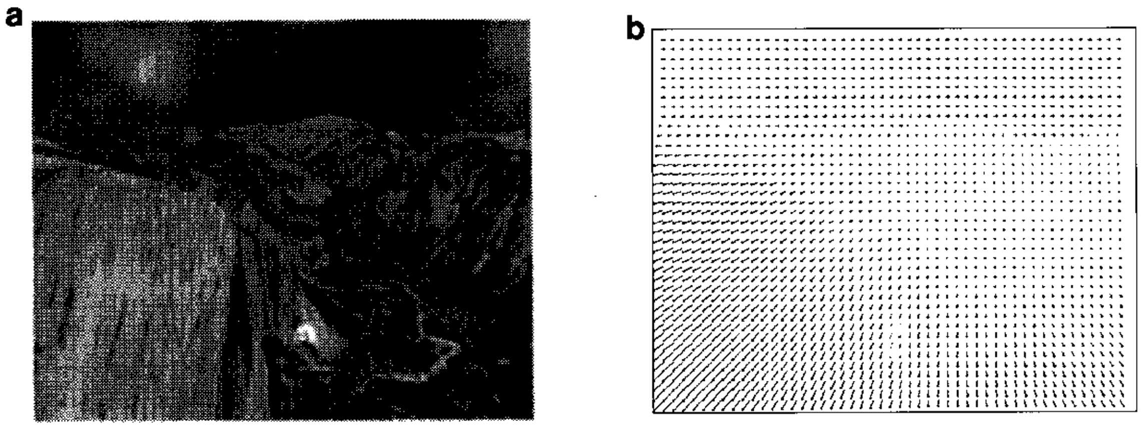

FIG. 8. (a) Yosemite fly-by, (b) Yosemite fly-by flow field.

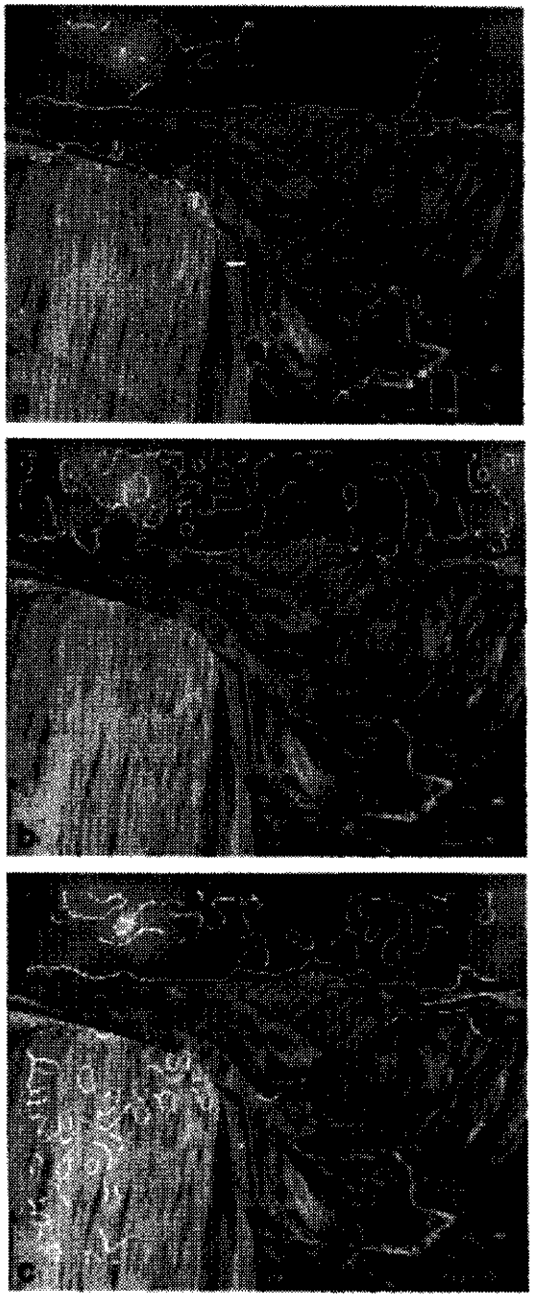

FIG. 9. (a) Our algorithm's result, (b) Schunck's result, (c) Thompson's result.

Z-transform, we derive

$$y(t) = \alpha_0 x(t) + \alpha_1 x(t-1) + \alpha_2 x(t-2) + \beta_1 y(t-1) + \beta_2 y(t-2) + \beta_3 y(t-3), \quad (16)$$

where

$$\alpha_0 = 1, \quad \alpha_1 = (-2 + b - b^2)e^{-b}, \quad \alpha_2 = (1 - b - b^2)e^{-2b}, \quad \beta_1 = -3e^{-b}, \quad \beta_2 = 3e^{-2b}, \quad \text{and} \quad \beta_3 = -e^{-3b}. \quad (17)$$

The filter is applied in both the x and y directions to find ridges of any orientation by thresholding on the maximum response of the two. The details of applying such recursive filters can be found in [39].

The major reason why we choose to use the recursive filter is that it is efficient. Canny's optimal ridge edge detector has a large filter support (because of the width of the ridge) and therefore the filtering is time-consuming. The second reason is that the recursive filter is flexible in the selection of multiple response criteria [39]; that is, we can easily select a different b in Eq. (15) to change the filter shape to match ridges of different sizes. The residual ridge size is dependent on the window size of the Hermite polynomial filter as analyzed in the residual profile. In the Canny approach, on the other hand, the optimal filter needs to be laboriously recomputed and generated. Our recursive filter, however, is not optimal for detecting ridge edges. This is a sacrifice we must make for real-time performance.

For the experiments described below, we use only the 3-D translation motion model with the first three linear equations in (12), and the Hermite polynomial filter size is $13 \times 13 \times 5$. The input image size is 64×64. The recursive ridge detector is a 2-D separable filter and (0.25, 0.5, 0.25) is the mask we use perpendicular to the ridge direction. The recursive function is designed so that the zero crossings are at ± 4, i.e., $b = 8/(\sqrt{5} + 1)$. The implementation runs at roughly 4 frames per second. The following figures illustrate the results. In these figures, the upper left image displays the scene, the lower left image displays the boundary map, and the image on the right displays the flow in needle map form.

Figure 12 illustrates the results of an arm moving in front of a stationary background. Note that the arm is a deformable object so the flow appears nonuniform. However, the boundary map captures the arm, despite the lack of texture in some portions of the arm.

Figure 13 illustrates the results of a head moving in front of a stationary background. The flow and boundary in this case both reveal the shape of the head in its entirety.

Figure 14 illustrates the results of boundary detection in the case of motion parallax. The scene contains two moving objects, a book in the foreground and a face in the background, and the camera is stationary. Although the two objects are physically moving at the same speed, their motions as perceived by the camera are different due to motion parallax. The result distinguishes the boundary of the book even though the motion parallax, as evidenced by the flow field, is small.

6. CONCLUSION

Motion boundary extraction algorithms are as important as motion estimation algorithms for the complete motion recovery problem. However, the interdependency of these two classes of algorithms poses a computational dilemma that renders any partial solution inaccurate. Indeed, the only way to solve the

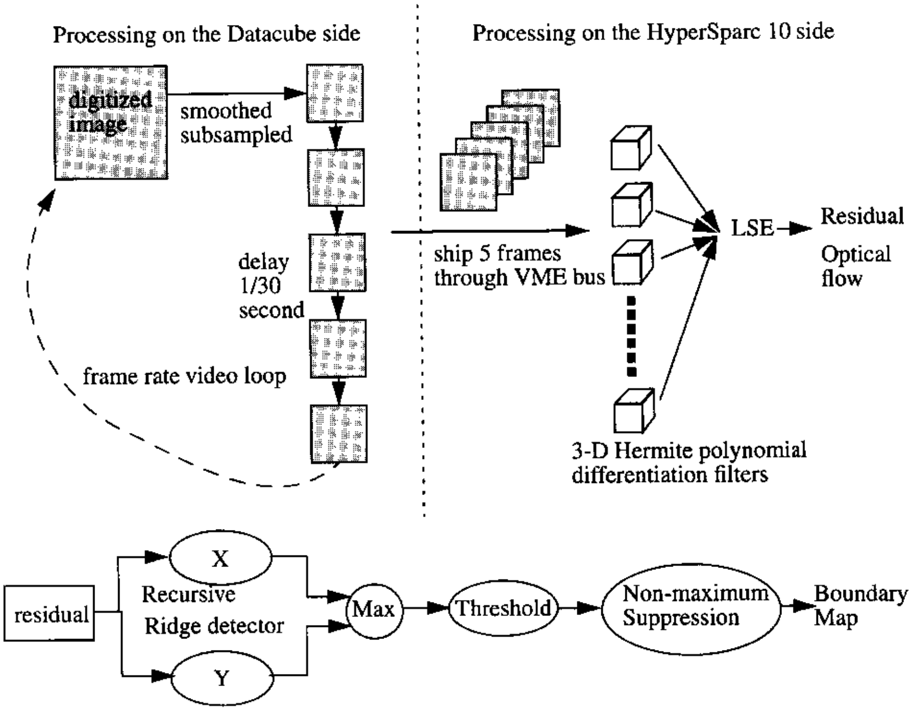

FIG. 10. Real-time implementation of our optical flow and boundary extraction algorithm.

motion recovery problem is to simultaneously address both motion segmentation and motion estimation. While recent research has focused on iterative methods, we propose a method based on a general motion model. This method is local, noniterative, and simultaneously deals with both motion estimation and motion 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 when we inspect its profile in the spatial domain.

The distinctive advantage of our local approach is its speed. It is demonstrated by our real-time implementation. This is ultimately important for obstacle avoidance and navigation applications.

APPENDIX A

Let A and b, defined in (12), contain no noise, that is $N = 0$ and $\Delta b = 0$. Then each equation of the system should hold, so [43]

$$E = As + b = 0 \quad \text{and} \quad s = -(A^T A)^{-1} A^T b. \tag{18}$$

Let the noise-contaminated solution be $\tilde{s}$ and the new residual

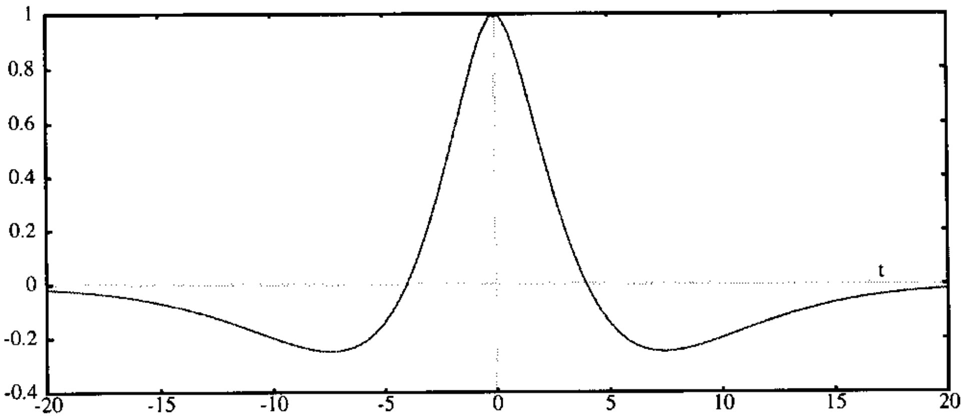

FIG. 11. Recursive ridge detector.

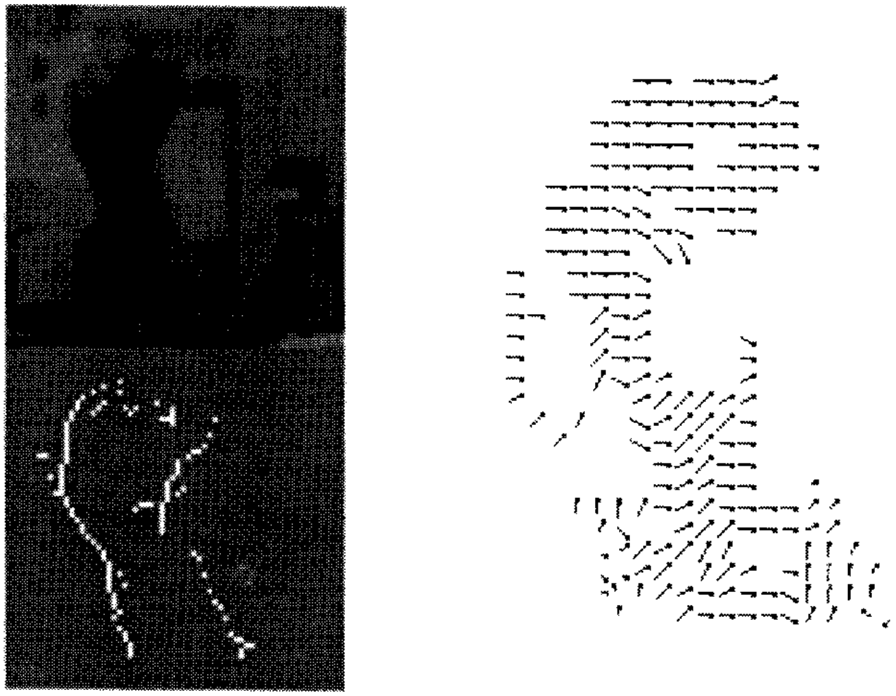

FIG. 12. Real-time flow and boundary result of a moving arm.

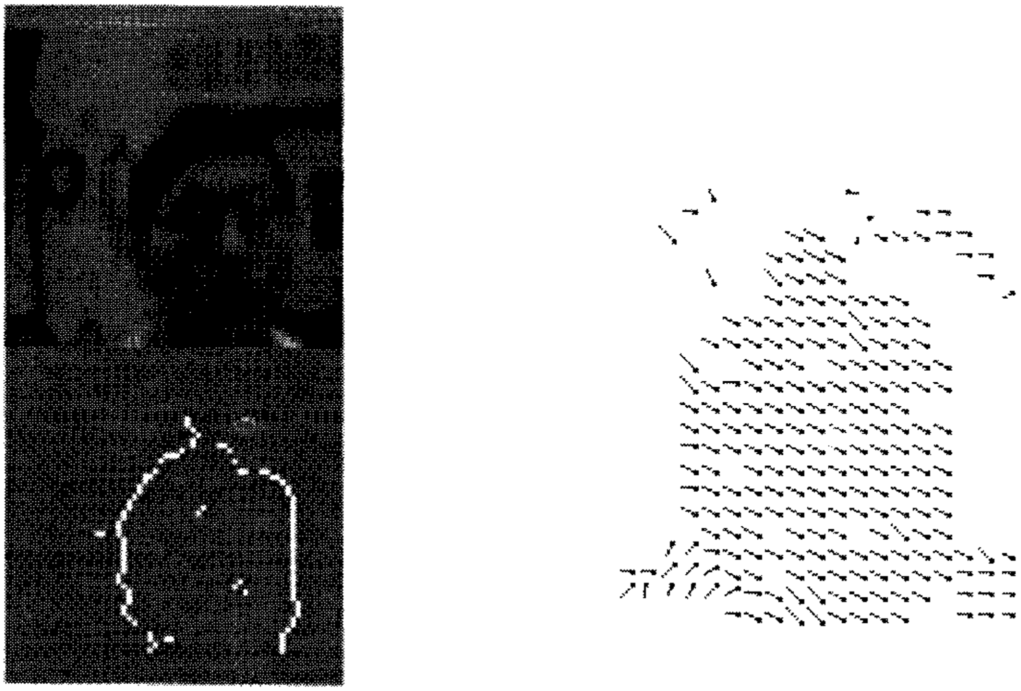

FIG. 13. Real-time flow and boundary result of a shaking head.

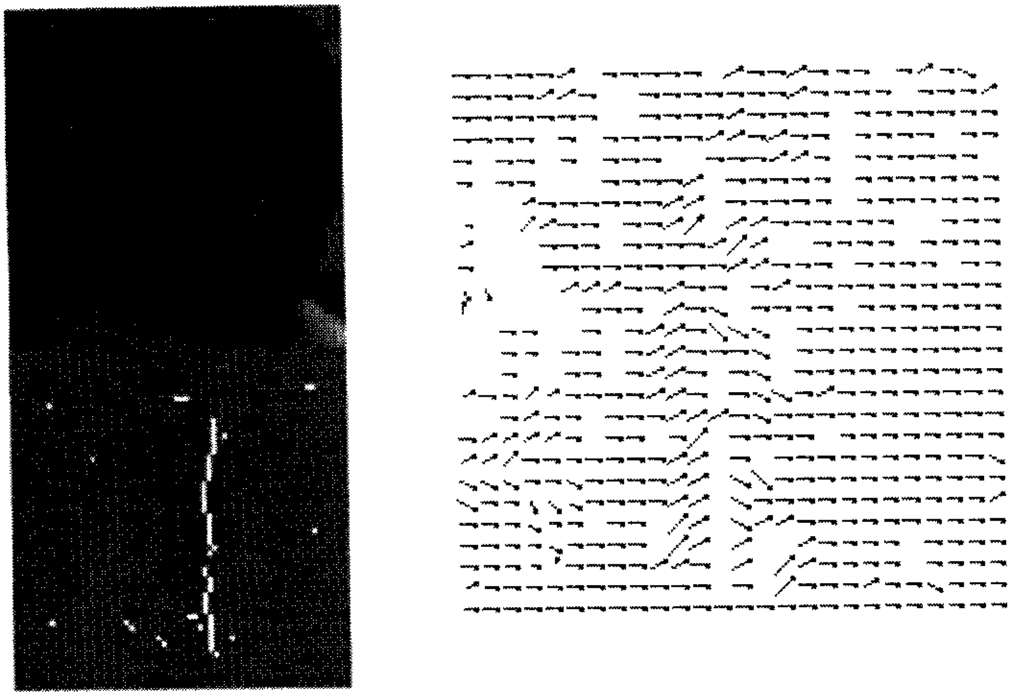

FIG. 14. Real-time flow and boundary result of motion parallax between a book and a face.

be $\tilde{E}$, and assume that $N \ll A$ and $\Delta b \ll b$ elementwise, i.e., $NN^T \approx 0$ and $N\Delta b \approx 0$. Then

$$\tilde{s} = -[(A+N)^T(A+N)]^{-1}\,(A+N)^T(b+\Delta b), \text{ and} \tag{19}$$

$$\begin{aligned}[(A+N)^T(A+N)]^{-1} &\approx (A^TA[I+(A^TA)^{-1}(A^TN+N^TA)])^{-1}\\ &\approx [I-(A^TA)^{-1}(A^TN+N^TA)](A^TA)^{-1}, \text{ so}\end{aligned} \tag{20}$$

$$\begin{aligned}\tilde{s} \approx{}& -(A^TA)^{-1}A^Tb + (A^TA)^{-1}(A^TN+N^TA)(A^TA)^{-1}A^Tb\\ &-(A^TA)^{-1}N^Tb - (A^TA)^{-1}A^T\Delta b\end{aligned} \tag{21}$$

Using (18), this can be simplified as

$$\begin{aligned}\tilde{s} &\approx s - (A^TA)^{-1}A^TNs - (A^TA)^{-1}A^T\Delta b \quad \text{and}\\ \Delta s &\approx -(A^TA)^{-1}A^TNs - (A^TA)^{-1}A^T\Delta b.\end{aligned} \tag{22}$$

For the residual, substituting $\tilde{s}$ into (13) and using (18), we derive

$$\begin{aligned}\tilde{E} &\approx \|(A+N)s - A(A^TA)^{-1}A^TNs\\ &\qquad -A(A^TA)^{-1}A^T\Delta b + b + \Delta b\|\\ &\approx \|(I - A(A^TA)^{-1}A^T)(Ns+\Delta b)\|\\ &\approx \|I - A(A^TA)^{-1}A^T\|\|Ns+\Delta b\|.\end{aligned} \tag{23}$$

ACKNOWLEDGMENT

The authors 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. The real-time display of the flow needle map is provided by Dr. Ted Camus at National Institute of Standards and Technology.

REFERENCES

1. I. E. Abdou and W. K. Pratt, Quantitative design and evaluation of enhancement thresholding edge detectors, *Proc. IEEE* **67**(5), 1979.
2. G. Adiv, Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field, *IEEE Trans. Pattern Anal. Machine Intell.* **11**(5), 1989, 477–489.
3. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, Performance of optical flow techniques, *Int. J. Comput. Vision* **12**(1), 1994, 43–77.
4. J. Bergen, P. Burt, R. Hingorani, and S. Peleg, A three-frame algorithm for estimating two-component image motion, *IEEE Trans. Pattern Anal. Machine Intell.* **14**(9), 1992, 886–896.
5. M. Black and P. Anadan, The robust estimation of multiple motions: parametric and piecewise-smooth flow field, *Comput. Vision Image Understanding* **63**(1), 1996.
6. P. Bouthemy, A maximum-likelihood framework for determining moving edges, *IEEE Trans. Pattern Anal. Machine Intell.* **11**(5), 1989.
7. O. J. Braddick, A short-range process in apparent motion, *Vision Res.* **14**, 1974, 519–527.
8. J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Machine Intell.* **8**(11), 1986, 679–698.
9. T. Darrel and A. Pentland, Robust estimation of a multi-layered motion representation, in *Proceedings of IEEE Workshop on Visual Motion, Princeton, NJ, 1991*, pp. 173–178.

10. L. Davis, Z. Wu, and H. Sun, Contour-based motion estimation, University of Maryland TR-1179, June 1982.

11. C. Debrunner and N. Ahuja, Motion and structure factorization and segmentation of long multiple motion image sequences, *Proceedings of the IEEE European Conference on Computer Vision, Santa Margherita Ligure, Italy, 1992*, pp. 217–221.

12. J. Dengler, Estimation of discontinuous displacement vector fields with the minimum description length criterion, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lahaina, HI, 1991*, pp. 276–282.

13. D. J. Fleet and A. L. Jepson, Computation of component image velocity from local phase information, *Int. J. Comput. Vision* **5**(1), 1990, 77–104.

14. C. Giardina and E. R. Dougherty, *Morphological Methods in Image and Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1988.

15. N. M. Grzywacz, and A. L. Yuille, A model for estimate of local image velocity by cells in the visual cortex, *Proc. Roy. Soc. London, A* **239**, 1990, 129–161.

16. R. Hartley, Segmentation of optical flow fields by pyramid linking, *Pattern Recognit. Lett.* 3(5), 1985, 253–262.

17. M. Hashimoto and J. Sklansky, Multiple-order derivatives for detecting local image characteristics, *Comput. Vision Graphics Image Process.* **39**, 1987, 28–55.

18. D. J. Heeger, Optical flow using spatiotemporal filters, *Int. J. Comput. Vision* **1**(4), 1988, 279–302.

19. F. Heitz and P. Bouthemy, Multimodal estimation of discontinuous optical flow using Markov random fields, *IEEE Trans. Pattern Anal. Machine Intell.* **15**(12), 1993, 1217–1232.

20. F. Heyden, Evaluation of edge detection algorithms, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1992*, pp. 618–622.

21. E. Hildreth, *The Measurement of Visual Motion*, MIT Press, Cambridge, MA, 1984.

22. B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artificial Intell.* **17**, 1981, 185–204.

23. A. Jepson and M. Black, Mixture model for optical flow computation, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, 1993*, pp. 760–761.

24. J. K. Kearney, W. B. Thompson, and D. L. Boley, Optical flow estimation: An error analysis of gradient based methods with local optimization, *IEEE Trans. Pattern Anal. Machine Intell.* **9**, 1987, 229–244.

25. C. Koch, J. Marroquin, and A. Yuille, Analog "neural" networks in early vision, *Proc. Nat. Acad. Sci.* **83**, 1992, 4263–4267.

26. J. Konrad and E. Dubois, Bayesian estimation of motion vector fields, *IEEE Trans. Pattern Anal. Machine Intell.* **14**(9), 1992.

27. H. Liu, T. Hong, M. Herman, and R. Chellappa, *A Reliable Optical Flow Algorithm Using 3-D Hermite Polynomials*, NIST IR-5333, Dec. 1993.

28. H. Liu, T. Hong, M. Herman, and R. Chellappa, *A General Motion Model and Spatio-temporal Filters for Computing Optical Flow*, University of Maryland-TR -3365, NIST-IR 5539, Nov. 1994 [Also, *Int. J. of Comput. Vision*, to appear]

29. H. Liu, T. Hong, M. Herman, and R. Chellappa, A generalized motion model for estimating optical flow using 3-D Hermite polynomials, *Proceedings of the International Conference on Pattern Recognition, Jerusalem, Israel, 1994*, pp. 360–366.

30. H. Liu, T. Hong, M. Herman, and R. Chellappa, *Motion-Model-Based Boundary Extraction*, University of Maryland TR-3414, Feb. 1995. [Also, *Proceedings of IEEE International Symposium on Computer Vision, Coral Gables, FL, Nov. 1995*, pp. 587–592].

31. H. Liu, *A General Motion Model and Spatio-temporal Filters for 3-D Motion Interpretations*, Ph.D. dissertation, University of Maryland, Sept. 1995.

32. D. Murray and B. Buxton, Scene segmentation from visual motion using global optimization, *IEEE Trans. Pattern Anal. Machine Intell.* **9**(2), 1987, 220–228.

33. D. Murray and N. S. Williams, Detecting the image boundaries between optical flow fields from several moving planar facets, *Pattern Recognit. Lett.* 4(2), 1986, 87–92.

34. K. Mutch and W. Thompson, Analysis of accretion and deletion at boundaries in dynamic scenes, *IEEE Trans. Pattern Anal. Machine Intell.* 7(2), 1985, 133–138.

35. H. H. Nagel, Direct estimation of optical flow and of its properties, in *Artificial and Biological Vision Systems* (G. A. Orban and H. H. Nagel, Eds.), 1974, pp. 193–224.

36. K. Nakayama and J. M. Loomis, Optical velocity patterns, velocity sensitive neurons, and space perception: A hypothesis, *Perception* **3**, 1974.

37. R. Nelson and Y. Aloimonos, Obstacle avoidance using flow field divergence, *IEEE Trans. Pattern Anal. Machine Intell.* **11**(10), 1989, 1102–1106.

38. J. L. Potter, Velocity as a cue to segmentation using motion information, *IEEE Trans. Systems, Man, Cybern.* **5**, 1975, 390–395.

39. S. Sarkar and K. L. Boyer, On optimal infinite impulse response ridge detection filters, *IEEE Trans. Pattern Anal. Machine Intell.* **13**(11), 1991, pp. 1154–1171.

40. B. Schunck, Image flow segmentation and estimation by constraint line clustering, *IEEE Trans. Pattern Anal. Machine Intell.* **11**(10), 1989, 1010–1027.

41. M. Shizawa and K. Mase, A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lahaina, HI, 1991*, pp. 289–295.

42. A. Spoerri and S. Ullman, *The Early Detection of Motion Boundaries*, A. I. Memo No. 935, AI Lab, MIT, 1987.

43. G. W. Stewart, *Introduction to Matrix Computation*, Academic Press, New York, 1973.

44. W. Thompson, K. Mutch, and V. Berzins, Dynamic occlusion analysis in optical flow fields, *IEEE Trans. Pattern Anal. Machine Intell.* **7**(4), 1985, 374–383.

45. A. Verri and T. Poggio, Motion field and optical flow: Qualitative properties, *IEEE Trans. Pattern Anal. Machine Intell.* **11**(5), 1989, 490–498.

46. P. Werkhoven, and J. J. Koenderink, Extraction of motion parallax structure in the visual system I, *Biol. Cybern.* **63**, 1990, 185–191.

47. R. A. Young, Simulation of human retinal function with the Gaussian derivative model, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 1986*, pp. 564–569.