Optic Flow Field Segmentation and Motion Estimation Using a Robust Genetic Partitioning Algorithm

Yan Huang, *Member, IEEE,* Kannappan Palaniappan, *Member, IEEE,*
Xinhua Zhuang, *Senior Member, IEEE,* and Joseph E. Cavanaugh

***Abstract*—Optic flow motion analysis represents an important family of visual information processing techniques in computer vision. Segmenting an optic flow field into coherent motion groups and estimating each underlying motion is a very challenging task when the optic flow field is projected from a scene of several independently moving objects. The problem is further complicated if the optic flow data are noisy and partially incorrect. In this paper, we present a novel framework for determining such optic flow fields by combining the conventional robust estimation with a modified genetic algorithm. The baseline model used in the development is a linear optic flow motion algorithm [38] due to its computational simplicity. The statistical properties of the generalized linear regression (GLR) model are thoroughly explored and the sensitivity of the motion estimates toward data noise is quantitatively established. Conventional robust estimators are then incorporated into the linear regression model to suppress a small percentage of gross data errors or *outliers*. However, segmenting an optic flow field consisting of a large portion of incorrect data or multiple motion groups requires a very high robustness that is unattainable by the conventional robust estimators. To solve this problem, we propose a genetic partitioning algorithm that elegantly combines the robust estimation with the genetic algorithm by a bridging genetic operator called *self-adaptation*.**

***Index Terms*—motion estimation, optic flow field segmentation, linear regression, robust estimation, genetic algorithm.**

I. Introduction

An optic flow field results from two successive image frames in time-varying image sequences that measure the projection of 3-D scenes. The notion of optic flow field or displacement field is used in a variety of applications, notably in rigid motion analysis [14], [27], [28], [38], two-view matching [32], and image sequence coding [7]. At a preliminary level, the optic flow motion analysis is to detect or flag motion(s) in the viewed scene. At a more sophisticated level, it is to infer the motion parameters and surface structures of moving object(s). The optic flow motion analysis basically has two related stages. The first stage involves the computation of the optic flow field from monocular image sequences. The second stage is to meaningfully interpret the computed optic flow field regarding the underlying object structures and motion parameters. There is a large body of literature on optic flow motion analysis. For a comprehensive overview, the readers are referred to an excellent review paper by Aggarwal and Nandhakumar [3].

Many previous research works are confined to the case where the scene contains one single 3-D rigid body motion. Along this direction, several researchers have developed nonlinear equations for recovering the 3-D motion and structure [4], [24], [34] that are solved by iterative numerical methods. However, this approach does not guarantee unique solution due to the initial-guess dependence. Longuet-Higgins [19] and Tsai and Huang [27] are among the first to propose computationally simpler techniques based on linear equations. The linear algorithm is subsequently unified by Zhuang and Faugeras [37] and simplified by Zhuang, Huang, and Haralick [38]. However, linear algorithms are reported to be sensitive to noises [27], [33]. This is our initial motivation for developing robust algorithms to cope with this problem.

The nature of the optic flow motion analysis has also been studied. It is well known that this inverse problem is ill-posed in that the observed data do not uniquely constrain the solution. Thus, various smoothness constraints are proposed to remedy this problem [13], [23], [25]. The inherent ambiguities in recovering the 3-D motion are also investigated [2], [35]. These adversary aspects have raised some concerns about the suitability of quantitative 3-D motion estimation from optic flow fields [30]. However, the debate on this topic is beyond the scope of this paper.

Meanwhile, a more important yet challenging task, i.e., the determination of optic flow fields containing multiple independently moving objects, has received much attention. Thompson [26] presents a segmentation scheme that combines the contrast information with optic flow field and uses a region growing method to merge similar velocity patches. Adiv [1] develops a grouping technique based on a computationally expensive Hough voting scheme which recovers the 3-D motions and structures using verified object hypotheses. The optic flow field is allowed to be sparse, noisy and partly incorrect since the Hough transform is relatively insensitive to noises and partially incorrect data. Recently, Markov random fields are utilized to model dense optic flow fields [11], [21] and the segmentation corresponds to a global optimal solution. Darrell and Pentland

Manuscript received Dec. 13, 1994; revised July 10, 1995.

Y. Huang is with AT&T Bell Laboratories, Middletown, NJ 07748; e-mail: yan@mtdcr.att.com.

K. Palaniappan is with the University Space Research Association, Laboratory for Atmospheres, NASA Goddard Space Flight Center, Greenbelt, MD 20771.

X. Zhuang is with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211; e-mail: zhuang@ece.missouri.edu.

J.E. Cavanauagh is with the Department of Statistics, University of Missouri, Columbia, MO 65211.

To order reprints of this article, e-mail: transactions@computer.org, and reference IEEECS Log Number P95142.

[5] apply image constraints to each model in a multimodel regularization network. Wang and Anderson [31] combine robust estimation with k-mean clustering to achieve multiple motion segmentation. Some alternative approaches that make use of stereo images pairs or 3-D data have been attempted more recently. Chen and Huang [5] present a two-stage algorithm that first matches 3-D line segments and then infers the underlying multiple objects. Zhang and Faugeras [36] develop a hypothesize-and-verify paradigm to obtain the displacement field from two 3-D frames. Zhuang and Huang [40] use the MF-estimator [39] for robust 3-D to 3-D multiple pose estimation.

In this paper, we propose a novel framework for determining the optic flow field projected by single or multiple 3-D rigid body motions with noisy and partially incorrect data. Our original motivation is to extend the linear optic flow motion algorithm [38] by using robust estimation to cope with partially incorrect data sets. That is, the data sets contain a portion of *outliers* which significantly conflict with the underlying models or structures. Many previous works heavily rely on the availability of reliable optic flow data. Unfortunately, the task of acquiring reliable optic flow data is nontrivial. For example, for monocular image sequences, a matching procedure is usually used to establish the correspondence of image pixels to obtain the optic flow field. Thus, the quality of the computed optic flow field is usually not guaranteed as the matching procedure may contribute many outliers [10]. In addition, sensor errors and calibration errors can also cause outliers. As known, the presence of outliers in motion analysis is disastrous since it greatly affects the estimation accuracy. Thus, one realistic solution is to incorporate robust estimation methods into the interpretation process so that it can suppress the influence of outliers. In other words, we want to segment the optic flow data into a coherent motion group and a portion of outliers, then estimate the motion using only the relevant data. It is increasingly realized that robust methods are vital to all computer vision problems [20], [39], [41].

When the optic flow field is projected from several independently moving objects, segmenting it into multiple coherent motion groups and estimating each associated motion is a more challenging task. Interesting enough, however, we find that it can also be treated as a robust estimation problem. This is because with respective to one particular motion group, all other motion groups and the outliers appear as a larger set of outliers, all of which conflict with the first motion group. Once a consistent motion group is extracted by the robust algorithm, the same procedure can be applied on the remaining portion of the data set to further extract other motion groups in a recursive fashion. However, since the outlier portion may now be very high with respect to each motion group, we need a robust estimator possessing a high degree of robustness. That is, it must be able to extract a consistent motion group without much influenced by a large portion of other irrelevant data. For example, to handle an optic flow field equally divided over three motion groups, the robust estimator needs an outlier tolerance of at least 67%. Unfortunately, the conventional robust estimators, such as Huber's M-estimator [16], have a typical outlier tolerance of 5% to 10%, which is far from satisfactory.

In the paper, we propose a robust genetic partitioning algorithm to achieve the desired high robustness by establishing a close connection between the robust estimation procedure and the genetic algorithm [9], [12]. In our algorithm, the optic flow field is binary partitioned and encoded as chromosomes such that each chromosome ensemble corresponds to one motion group. The algorithm alternates between a segmentation stage and an estimation stage. The segmentation stage is governed by a modified genetic algorithm which starts with a set of random partitions and seeks increasingly consistent partition(s). The estimation stage uses a robust estimator to decide the current motion model and reject respective outliers. These two stages are bridged by a newly defined genetic operator called *self-adaptation*, which updates each partition based on the motion estimates. This operator tremendously speeds up the otherwise slow genetic algorithm. The fitness function of each partition is determined by two sources, i.e., its compactness measure and its least-squares fitting error to the linear optic flow motion equation. In a departure from some previous works, our algorithm does not require spatial connectivity of a motion group and allows for sparse optic flow fields. This is important since, besides being a computational burden, acquiring dense optic flow fields is sometimes difficult if the feature points are only sparsely available [27], [29].

The paper is organized as follows. Section II provides a brief overview of the linear optic flow motion algorithm. In Section III, the statistical properties of the generalized linear regression (GLR) model are thoroughly explored and the sensitivity of the motion estimates toward data noise is derived. In Section IV, two conventional residual-based robust estimators, namely, the Huber's M-estimator and Tukey's biweight estimator are incorporated into the linear regression model. Section V provides the skeletons of the basic genetic algorithm and then presents our robust genetic partitioning algorithm. Section VI shows the experimental results. Section VII is a conclusion.

II. Linear Optic Flow Motion Estimation

The optic flow motion analysis concerns with the perspective projection of 3-D rigid body motions onto a 2-D image domain. Suppose that a rigid body motion takes place in the half 3-D space in front of a camera, i.e., $z < 0$. Let $\boldsymbol{p}(t) = (x(t), y(t), z(t))^T$ be an object surface point at time t. The instantaneous 3-D rigid motion is represented by

$$\dot{\boldsymbol{p}}(t) = \boldsymbol{\omega}(t) \times \boldsymbol{p}(t) + \boldsymbol{k}(t), \tag{1}$$

where the overhead dot represents the time derivative, "$\times$" the cross product, $\boldsymbol{\omega}(t) = (\omega_1(t), \omega_2(t), \omega_3(t))^T$ the instantaneous rotational angular velocity, and $\boldsymbol{k}(t) = (k_1(t), k_2(t), k_3(t))^T$ the instantaneous translational velocity. Let $(X(t), Y(t))$ be the 2-D central perspective projection of $\boldsymbol{p}(t)$ onto image plane $z = 1$. Let $(u(t), v(t)) = (\dot{X}(t), \dot{Y}(t))$ denote the projective instantaneous velocity of $(X(t), Y(t))$. The perspective projection establishes the following relationships:

$$\begin{aligned} X(t) &= x(t)/z(t), \\ Y(t) &= y(t)/z(t), \\ \boldsymbol{p}(t) &= z(t)(X(t), Y(t), 1)^T, \end{aligned} \tag{2}$$

and

$$u(t)=\omega_2(t)-Y(t)\omega_3(t)-\{\omega_1(t)Y(t)-\omega_2(t)X(t)\}X(t) +\{k_1(t)-k_3(t)X(t)\}/z(t),$$
$$v(t)=\omega_3(t)X(t)-\omega_1(t)-\{\omega_1(t)Y(t)-\omega_2(t)X(t)\}Y(t) +\{k_2(t)-k_3(t)Y(t)\}/z(t). \quad (3)$$

The element $\{(X(t), Y(t)), (u(t), v(t))\}$ is called an *optic flow image point*. The optic flow motion analysis is to estimate the motion parameters ω and $\boldsymbol{k}$, given a finite set of observed optic flow image points $\{(X_i, Y_i), (u_i, v_i)\}$, $i = 1, \cdots, N$, at time t (the time indices are omitted for simplicity). The optic flow data set size N can be equal to or smaller than the number of image pixels depending on whether dense or (subsampled) sparse optic flow field is being considered. Given a pixel location (X_i, Y_i), the velocity (u_i, v_i) is usually computed by using optic flow algorithms [13], [3].

It is shown in Zhuang et al. [38] that the motion parameters can be obtained by solving $\boldsymbol{h}$ from the following set of homogeneous linear equations:

$$\tilde{\boldsymbol{a}}_i^T\boldsymbol{h}=0, \quad \text{for} \quad i=1,\cdots,N, \quad (4)$$

where

$$\tilde{\boldsymbol{a}}_i=(1, X_i^2, Y_i^2, 2X_iY_i, 2X_i, 2Y_i, -v_i, u_iX_i, v_iY_i, v_iX_i-u_iY_i)^T$$
$$\boldsymbol{h}=(h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8)^T.$$

The motion parameters ω and $\boldsymbol{k}$ can then be computed from $\boldsymbol{h}$ as shown below. It is justified in [38] that at least eight optic flow image points ($N \geq 8$) are needed to uniquely recover the motion parameters $(\omega, \boldsymbol{k})$ when $\boldsymbol{k}$ is nonzero, and that $\boldsymbol{k}$ can only be determined up to a constant scale. In other words, we can only estimate the motion parameters by $(\hat{\omega}, \hat{\boldsymbol{k}})$ with $\hat{\boldsymbol{k}}$ being colinear to $\boldsymbol{k}$. The depth information, i.e., z_i, $i = 1, \cdots, N$ can also be recovered once $(\hat{\omega}, \hat{\boldsymbol{k}})$ is obtained. The following is the procedure to compute $\hat{\omega}$ and $\hat{\boldsymbol{k}}$ from the estimated $\hat{\boldsymbol{h}}$ (see [38] for the details).

Step 1. Set $\hat{\boldsymbol{k}}=(\hat{h}_6, \hat{h}_7, \hat{h}_8)^T$.

Step 2. If $(|\hat{h}_6| \geq |\hat{h}_7|)$ and $(|\hat{h}_6| \geq |\hat{h}_8|)$, then

$\hat{\omega}_1=(\hat{h}_1-\hat{h}_2-\hat{h}_0)/(2\hat{h}_6)$, $\hat{\omega}_2=(2\hat{h}_3-\hat{h}_7*\hat{\omega}_1)/\hat{h}_6$, and $\hat{\omega}_3=(2\hat{h}_4-\hat{h}_8*\hat{\omega}_1/\hat{h}_6$.

Step 3. Otherwise, if $(|\hat{h}_7| \geq |\hat{h}_8|)$, then

$\hat{\omega}_2=(\hat{h}_2-\hat{h}_0-\hat{h}_1)/(2\hat{h}_7)$, $\hat{\omega}_1=(2\hat{h}_5-\hat{h}_8*\hat{\omega}_2)/\hat{h}_7$, and $\hat{\omega}_3=(2\hat{h}_3-\hat{h}_6*\hat{\omega}_2)/\hat{h}_7$.

Step 4. Otherwise,

$\hat{\omega}_3=(\hat{h}_0-\hat{h}_1-\hat{h}_2)/(2\hat{h}_8)$, $\hat{\omega}_1=(2\hat{h}_4-\hat{h}_6*\hat{\omega}_3/\hat{h}_8$, and $\hat{\omega}_3=(2\hat{h}_5-\hat{h}_7*\hat{\omega}_3)/\hat{h}_8$.

To suppress the pixel location factor, the linear optic flow motion equations can be normalized as follows,

$$\boldsymbol{a}_i^T\boldsymbol{h}=0, \quad \text{for} \quad i=1,\cdots,N, \quad (5)$$

where $\boldsymbol{a}_i=\frac{1}{\sqrt{X_i^2+Y_i^2}}\tilde{\boldsymbol{a}}_i$, or in matrix form,

$$\boldsymbol{A}\boldsymbol{h}=\boldsymbol{o}, \quad (6)$$

with $\boldsymbol{A}=(\boldsymbol{a}_1^T, \boldsymbol{a}_2^T, \cdots, \boldsymbol{a}_N^T)^T$. In practice, the observed optic flow image points $\{(X_i, Y_i), (u_i, v_i)\}$ may contain noises. Since the velocity component (u_i, v_i) is usually obtained from the displacement of (X_i, Y_i) between two consecutive image frames, the noises in (u_i, v_i) and (X_i, Y_i) are relative. Hence, we may assume that only the velocity components (u_i, v_i) carry noises. Thus, according to (5) only the last three components in $\boldsymbol{a}_i$ may carry small noises that are modeled as Gaussian noises. In the next section, we will present the solution and the statistical properties of (5) under Gaussian noise condition while exploring a more geneal linear regression model. In Section IV, we will incorporate robust estimation methods to cope with possible outliers in $\{(X_i, Y_i), (u_i, v_i)\}$.

III. General Linear Regression Model

A. Multiple Linear Regression Model

The multiple linear regression (MLR) model is well studied in statistical literature [22]. It is represented by

$$y_i=\beta_0+\beta_1x_{1i}+\beta_2x_{2i}+\cdots+\beta_px_{pi}+\epsilon_i, \quad i=1,\cdots,N, \quad (7)$$

where x_{ji} are deterministic measurement variables, and y_i are stochastic response variables, or in matrix form

$$\boldsymbol{y}=\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\epsilon}, \quad (8)$$

where $\boldsymbol{y}=(y_1, \cdots, y_N)^T$, $\boldsymbol{\beta}=(\beta_0, \cdots, \beta_p)^T$, $\boldsymbol{\epsilon}=(\epsilon_1, \cdots, \epsilon_N)^T$, and $\boldsymbol{X}$ is a *design matrix*

$$\boldsymbol{X}=\begin{pmatrix}1 & x_{11} & x_{21} & \cdots & x_{p1}\\ 1 & x_{12} & x_{22} & \cdots & x_{p2}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & x_{1N} & x_{2N} & \cdots & x_{pN}\end{pmatrix}. \quad (9)$$

Under the assumption that $\boldsymbol{\epsilon}$ follows normal distribution $N(\boldsymbol{o}, \sigma^2\boldsymbol{I})$, and the design matrix has full column rank, the least-squares solution to (8) is obtained as $\hat{\boldsymbol{\beta}}=(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$ [22]. The MLR model finds many applications in economics and social sciences. But it has a serious limitation that the measurement variables are deterministic. Thus, the so-called "measurement-in-error" MLR model, which allows for random measurement variables, has been investigated under certain asymptotic assumptions [8].

For our purpose, the foregoing models cannot be directly utilized since the normalized optic flow linear (5) is homogeneous and thus possesses different statistical properties. Therefore, an effort is made to investigate the statistical properties of the homogeneous linear equation in a general setting, called the *general linear regression* (GLR) represented by

$$\beta_0+\beta_1x_{1i}+\beta_2x_{2i}+\cdots+\beta_px_{pi}=\epsilon_i, \quad i=1,\cdots,N, \quad (10)$$

where x_{ji} are normally distributed random variables, i.e., $x_{ji} \sim N(\bar{x}_{ji}, \sigma_j^2)$, β_j are linear coefficients, and ϵ_i are mutually independent random variables (but dependent on the x_{ji}) measuring the a priori model errors. A general assumption for x_{ji} is that $\sigma_{jk} \overset{def}{=} cov(x_{ji}, x_{ki}) \neq 0$ for all j, k, and $cov(x_{ji}, x_{jk}) = 0$ for all $i \neq k$. The matrix form of (10) is

$$X\beta = \epsilon. \tag{11}$$

For the sake of notational clarity, we should point out that X and β correpond to A and h in the previous section, respectively. To facilitate the development, we first establish the corresponding noise-free model as

$$\beta_0 + \beta_1 \bar{x}_{1i} + \beta_2 \bar{x}_{2i} + \cdots + \beta_p \bar{x}_{pi} = 0, \quad i = 1, \cdots, N, \tag{12}$$

or in matrix form

$$\bar{X}\beta = \mathbf{0}, \tag{13}$$

where $\bar{X}$ denotes the *mean* design matrix. It follows from (12) that the mean design matrix is 1-degree degenerated, namely,

$$rank(\bar{X}^T \bar{X}) = p. \tag{14}$$

Due to the homogeneity of (12), we may assume without loss of generality that $\beta_0 = -1$ in the following development.

Let $\mathbf{1}$ denote a N-dimensional vector with each component being 1. Partition the design matrix as $X = [\mathbf{1} \mid X^*]$, the mean design matrix as $\bar{X} = [\mathbf{1} \mid \bar{X}^*]$, and the coefficient vector as $\beta^T = (-1, \beta^{*T})$. By (13), we have

$$-\mathbf{1} + \bar{X}^* \beta^* = \mathbf{0}. \tag{15}$$

Let $\Delta X^* \overset{def}{=} X^* - \bar{X}^*$. It follows from the foregoing noise assumption that each row of ΔX^* is independently distributed as $N(\mathbf{0}, \Sigma)$, where Σ is a covariance matrix with σ_{jk} being its jkth element. It is then straightforward to verify that $\epsilon \sim N(\mathbf{0}, \beta^{*T} \Sigma \beta^* I)$.

B. Least-Squares Estimation

The least-squares estimate $\hat{\beta}^* = (\hat{\beta}_1, \cdots, \hat{\beta}_p)^T$ of the GLR model (11) is determined by

$$\min_{\beta} \|X\beta\|^2 = \min_{\beta^*} \left\| -\mathbf{1} + X^* \beta^* \right\|^2 \tag{16}$$

which leads to

$$\hat{\beta}^* = \left(X^{*T} X^*\right)^{-1} X^{*T} \mathbf{1}. \tag{17}$$

Due to (15), we can verify that

$$\hat{\beta}^* = \beta^* - \left(X^{*T} X^*\right)^{-1} X^{*T} \Delta X^* \beta^*. \tag{18}$$

Therefore, β^* is a biased estimate since

$$E(\hat{\beta}^*) = \beta^* - \beta^* E\left((X^{*T} X^*)^{-1} X^{*T} \Delta X^*\right), \tag{19}$$

and the second term is usually nonzero. However, when the noise factor is small ($\Delta X^* \to \mathbf{0}$), $\hat{\beta}^*$ is asymptotically unbiased.

It is very difficult to analytically derive the variance of $\hat{\beta}^*$ because of the complicated random matrix operations in (17). However, by applying the Taylor expansion at $\bar{X}^*$, we may derive an approximation of $Var(\hat{\beta}^*)$ as follows. First,

$$\begin{aligned} \hat{\beta}^* &= \beta^* - \left(X^{*T} X^*\right)^{-1} X^{*T} \Delta X^* \beta^* \\ &= \beta^* - \left((\bar{X}^{*T} \bar{X}^*)^{-1} \bar{X}^{*T} \Delta X^* + o(\Delta X^*)\right) \beta^*. \end{aligned} \tag{20}$$

When the random factor ΔX^* is small as compared to the nonrandom factor $\bar{X}^*$, we may ignore the higher order terms $o(\Delta X^*)$ and obtain

$$Var(\hat{\beta}^*) \approx \beta^{*T} \Sigma \beta^* (X^{*T} X^*)^{-1}. \tag{21}$$

This equation characterizes the sensitivity of the estimate $\hat{\beta}^*$ with respect to the noise factor Σ. It explains the previously reported high noise sensitivity of the linear optic flow motion algorithms [27], [33]. That is, the variance of the linear estimate (and subsequently the motion parameters) is related to not only the noise covariance Σ but also the true motion parameters (reflected by β^* and the data location (reflected as $(X^{*T} X^*)^{-1}$). Thus, when any of these factors has large numerical values, the reliability of the motion estimates is reduced.

C. Residual Analysis

The residuals of the GLR model $e_1, \cdots, e_N$ are defined to be the (post) *model fitting errors*,

$$e_i = -1 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \cdots + \hat{\beta}_p x_{pi}, \quad i = 1, \cdots, N. \tag{22}$$

Denote $\boldsymbol{e} = (e_1, \cdots, e_N)^T$ and define the *hat matrix* $\boldsymbol{H}^* = X^* (X^{*T} X^*)^{-1} X^{*T}$. We can verify

$$\begin{aligned} \boldsymbol{e} &= -\mathbf{1} + X^* \hat{\beta}^* \\ &= (\boldsymbol{I} + \boldsymbol{H}^*) \epsilon, \end{aligned} \tag{23}$$

which meaningfully relates the post model residual $\boldsymbol{e}$ with the a priori model error ϵ. Since $\boldsymbol{H}^*$ is a random matrix, it is difficult to obtain the exact distribution of $\boldsymbol{e}$ from (23). However, by using a zero-order Taylor expansion, it can be verified that

$$Var(\boldsymbol{e}) \approx (\boldsymbol{I} - \boldsymbol{H}^*) \beta^{*T} \Sigma \beta^*. \tag{24}$$

In deriving it, we have used the property that $(\boldsymbol{I} - \boldsymbol{H}^*)$ is a projection matrix (both idempotent and symmetric). It follows that,

$$Var(\boldsymbol{e}_i) \approx (1 - h_{ii}^*) \beta^{*T} \Sigma \beta^*, \tag{25}$$

where h_{ii}^* is the ith diagonal element of $\boldsymbol{H}^*$. Thus, the distribution of e_i is related to h_{ii}^* which by definition is a distance measure of the data point x_i from the sample mean.

In conventional robust estimation methods, the residual is used for weight assignment such that the data points with large residuals receive small or zero weights and are thus treated as outliers. By (25), we see that the data location factor h_{ii}^* can significantly alter the information conveyed by the residual. Therefore, in employing residual-based robust estimators, the residual should be appropriately normalized to eliminate the data location factor:

$$e_i \leftarrow \frac{e_i}{\sqrt{1 - h_{ii}^*}}. \tag{26}$$

In the following, the notion of model residual always refers to the normalized one unless explicitly pointed out.

It is straightforward to apply these results to the linear optic flow motion (6) since it is a special case of the GLR model with the last three components of $\boldsymbol{h}$ being random variables.

Therefore, the least-squares motion estimate $\hat{\boldsymbol{h}}$ and the associated statistical properties are obtained. As well known, the least-squares estimation is extremely unreliable with the presence of outliers. Thus, we discuss robust estimation methods to cope with the outlier problem in the next section.

IV. Residual-Based Robust Estimators

Most robust methods utilize the residual information to identify outliers since a larger residual indicates larger deviation from the underlying model. In the following, two conventional robust estimators, namely, the Huber's M-estimator and Tukey's biweight estimator are presented and incorporated into the linear regression model.

A. Huber's M-Estimator

Suppose there is a location parameter μ to be estimated from a number of observations x_i, $i = 1, \cdots, N$, which come from the distribution $F(x - \mu)$. The principle of the M-estimator [16] is that the estimate $\hat{\mu}$ is chosen such that

$$\sum_{i=1}^{N} \rho(x_i - \mu) \tag{27}$$

is minimized, where $\rho(\cdot)$ is a differentiable continuous function. It can be shown that the function $\rho(e) = e^2$ yields the estimate $\hat{\mu}$ as the sample mean, $\rho(e) = |e|$ yields the estimate $\hat{\mu}$ as the sample median, and $\rho(e) = -\log f(e)$, with $f(e)$ being the density function of the random variable x, yields the maximum likelihood estimator. Let $e_i = x_i - \mu$. Define the *influence function* φ as the first derivative of ρ, i.e., $\varphi(\cdot) = \rho'(\cdot)$. Thus, minimizing (27) is equivalent to solving

$$\sum_{i=1}^{N} \varphi(e_i) = 0. \tag{28}$$

Huber's M-estimator [16] uses an influence function defined as

$$\varphi(e_i) = \begin{cases} e_i & |e_i| \le k \\ k & e_i \ge k \\ -k & e_i \le -k, \end{cases} \tag{29}$$

where k is a predefined constant. Since $\varphi(\cdot)$ is a nonlinear function, (28) usually needs to be solved by iterative numerical methods. It is easy to verify that solving (28) by using influence function (29) is equivalent to the following weighted least-squares minimization:

$$\min_{\mu} \sum_{i=1}^{N} w_i e_i^2, \tag{30}$$

where the weights are given as:

$$w_i = \begin{cases} 1 & |e_i| \le k \\ k / e_i & e_i \ge k \\ -k / e_i & e_i \le -k. \end{cases} \tag{31}$$

We now incorporate Huber's M-estimator into the GLR regression model. Based on the weighted least-squares principle (30), the Huber's M-estimator for the linear coefficient β^* in the GLR model is derived from

$$\begin{aligned} \min_{\beta^*} \sum_{i=1}^{N} w_i e_i^2 &= \min_{\beta^*} \left\| \boldsymbol{W}^{1/2} \boldsymbol{e} \right\|^2 \\ &= \min_{\beta^*} \left\| -\boldsymbol{W}^{1/2} \boldsymbol{1} + \boldsymbol{W}^{1/2} \boldsymbol{X}^* \beta^* \right\|^2, \end{aligned} \tag{32}$$

where $\boldsymbol{W}^{1/2} = \mathrm{diag}(w_1^{1/2}, \cdots, w_N^{1/2})$ is a weight matrix. Hence, we obtain the M-estimator of β^* as

$$\hat{\beta}^* = (\boldsymbol{X}^{*T} \boldsymbol{W} \boldsymbol{X}^*)^{-1} \boldsymbol{X}^{*T} \boldsymbol{W} \boldsymbol{1}, \tag{33}$$

where $\boldsymbol{W} = \boldsymbol{W}^{1/2} \boldsymbol{W}^{1/2} = \mathrm{diag}(w_1, \cdots, w_N)$. Notice that (33) and (31) form a recursive procedure to determine appropriate w_i and the linear coefficient $\hat{\beta}^*$. The recursion starts with an identity weight matrix $\boldsymbol{W}^0 = \boldsymbol{I}$. The convergence of the M-estimator has been proved by Huber [16].

B. Tukey's Biweight Estimator

The Tukey's biweight estimator is another type of the M-estimator. It defines the influence function as

$$\varphi(e_i) = \begin{cases} e_i (1 - e_i^2)^2 & |e_i| \le 1 \\ 0 & |e_i| > 1, \end{cases} \tag{34}$$

where e_i is scaled by

$$e_i \leftarrow \frac{e_i}{c \cdot e_{med}}, \tag{35}$$

with e_{med} being the median of the absolute residuals before scaling and c being a tuning constant (normally between 4 and 12). Hence, the corresponding weight function can be derived as

$$w_i = \begin{cases} (1 - e_i^2)^2 & |e_i| \le 1 \\ 0 & |e_i| > 1. \end{cases} \tag{36}$$

The biweight estimate of β^* is thus the same as in (33) except that the newly defined weight function (36) is substituted. Thus, a recursive biweight estimation procedure is formed by (33), (35), (36), starting with an identity weight matrix. This recursive procedure named as *robust estimation cycle* will be utilized in the estimation stage of the following robust genetic partitioning algorithm.

C. Discussions

In general, the residual-based weighting function is designed to reduce the influence of outliers in the parameter estimation. This is accomplished by assigning smaller weights to larger residuals. The M-estimator uses a predefined constant as a threshold. The biweight estimator uses an adaptive approach in that all weights are relatively assigned with respect to the median residual. The most important aspect of a robust estimator is the tolerance of outliers, or the *breakdown points*. In our linear regression application, the breakdown point is typically 5% to 10% for Huber's M-estimator and 30% to 40% for Tukey's biweight-estimator.

The foregoing robust estimation procedure is adopted so that we can recover a single 3-D rigid body motion from partially incorrect motion data. However, the breakdown points of these robust estimators are not high enough to handle the case where outlier data occupied the majority or multiple data structures

exist. To achieve the high robustness to handle such data sets, we combine the robust estimator with a genetic algorithm and propose the following robust genetic partitioning algorithm.

V. A Robust Genetic Partitioning Algorithm

The genetic algorithm is primarily used for global parameter optimization by a stochastic mechanism which imitates the biological systems of natural selection and gene reproduction [9], [12]. It possesses the power of exhaustive searching methods for finding global solutions while avoiding their prohibitive computational complexities.

A. Basic Genetic Algorithm

Each parameter being optimized is encoded as a finite-length binary string of 0s and 1s. The linear concatenation of the binary strings for all parameters composes a *chromosome*. This process represents the discretization of the parameter space where each chromosome corresponds to a possible solution. At any time, the algorithm maintains a finite set of chromosomes, called a *population* or *chromosome pool*. The chromosomes compete with each other to reproduce and survive. The chance for a chromosome to reproduce and survive is determined by its *fitness function*, which is defined based on specific applications. In a manner resembling the natural biological system development, the chromosomes are allowed to mate and mutate. Mating of a chromosome pairs by a crossover operation results in an offspring chromosome pair which carries the information of both parents. Mutation of a chromosome alters certain bits of the chromosome. In brief, a genetic algorithm has three basic operations: i.e., reproduction, crossover and mutation. The entire optimization process consists of a number of repeated applications of these operations, called *generations*. The algorithm stops after a fixed number of generations, or after the chromosome pool stabilizes.

The initial chromosome population can be randomly generated. Their fitness values are then calculated. Denote the fitness value of the ith chromosome Ξ_i by $f(\Xi_i), i = 1, \cdots, C$, where C is the population size. The chromosome Ξ_i gets reproduced with a probability $p_i(\Xi_i)$ proportional to its fitness value,

$$p_i(\Xi_i) = f(\Xi_i) / \sum_{j=1}^{C} f(\Xi_j). \tag{37}$$

By the reproduction operation, the original chromosomes are replaced by the reproduced chromosomes which compose a mating pool. Mating is then done by randomly pairing these chromosomes into couples, and performing crossover of each couple with a small predefined *crossover probability* P_c as follows. Suppose $\Xi = \xi_1\xi_2 \cdots \xi_N$ and $\Gamma = \gamma_1\gamma_2 \cdots \gamma_N$ are a selected chromosome pair where ξ_i and $\gamma_i \in \{0, 1\}$, $i = 1, \cdots, N$. A crossover position k is randomly determined between 1 and N. Then with probability P_c this chromosome pair generates their offsprings as $\Xi^* = \xi_1 \cdots \xi_k \gamma_{k+1} \cdots \gamma_N$ and $\Gamma^* = \gamma_1 \cdots \gamma_k \xi_{k+1} \cdots \xi_N$. Finally, mutation is performed by altering each bit of a chromosome with a very small probability P_m.

The reproduction operator ensures the succession of "good" solutions; the mating operator passes on "good" information to the offspring generation; the mutation operator creates new solutions and thus avoids local optima. Hence, the parameter space is searched in an random yet organized manner in approaching global optima while escaping local optima. Though the genetic algorithm utilizes only these simple operations, it has been shown to be a very effective method in many global optimization applications [9]. The mathematical aspect of the genetic algorithm is summarized by the *Schema Theorem* which is referred to Goldberg [9].

A major practical limitation of the basic genetic algorithm is its relatively slow convergence. In contrast, by introducing a new operator called *self-adaptation*, our modified genetic algorithm will have a much faster convergence rate.

B. Combining Robust Estimation and Genetic Algorithm

The robust estimation methods can be intimately combined with the genetic algorithm to achieve the desired high robustness.

B.1 Connection Between Robust Estimation and Genetic Algorithm

As mentioned, the goal of robust estimation is to exclude the influence of irrelevant data (outliers) in parameter estimation by assigning appropriate weights. In case of extracting one consistent motion group from a partially incorrect data set, each outlier should ideally be assigned a weight of 0, while non-outlier assigned a weight of 1. This forms an ideal binary segmentation of the data set. In case of extracting multiple motion groups from a mingled data set (which may also contain outliers), each consistent motion group also corresponds to a binary partition of the data set with each belonging data point assigned a weight of 1.

To associate the foregoing objective with the genetic algorithm, the key is the chromosome encoding. Since each chromosome is an ensemble of binary bits, it is natural to link it to the binary partition of the data set. That is, each chromosome is used to represent a binary partition of the data set with the chromosome length equal to the data set size N. The three genetic operations can then be used to seek proper partitions of the data set. From one perspective, we may view it as an optimization procedure performed in the parameter space of binary weights (which is very small as compared to the ordinary cases where all parameters need to be quantized in sufficient precision levels and searched). This procedure is called the *segmentation stage*. The other parameters (motion parameters h) are determined by the robust estimation cycle (33), (35), (36), instead of genetic searching. This procedure is called the *estimation stage*. The two procedures are bridged by a new genetic operator called *self-adaptation*.

B.2 Robust Estimation and Self-Adaptation

The self-adaptation operator is applied after the completion of the other three genetic operators. Let $\Xi = \xi_1\xi_2 \cdots \xi_N$ denote a submitted chromosome, which partitions the original data set $\{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_N\}$ (see (5)) into a relevant data subset $G(\Xi)$ and an irrelevant data subset $\overline{G(\Xi)}$ by

$$G(\Xi) = \{\boldsymbol{a}_{p_j} : \xi_{p_j} = 1,\ p_j \in \{1, \cdots, N\} \text{ and } j = 1, \cdots, N_1\}$$
$$\overline{G(\Xi)} = \{\boldsymbol{a}_{q_k} : \xi_{q_k} = 0,\ q_k \in \{1, \cdots, N\} \text{ and } k = 1, \cdots, N - N_1\} \quad (38)$$

where N_1 is the number of 1 bits in Ξ. Then, the relevant data subset $G(\Xi)$ is submitted to the *robust estimation cycle,* such as (33), (35), 36), for estimating the current motion model $\hat{\boldsymbol{h}}$ and computing the associated diagonal weight matrix $\boldsymbol{W} = \text{diag}\{w_{p_1}, \cdots, w_{p_{N_1}}\}$ where each $w_{p_j} (0 \le w_{p_j} \le 1)$ denotes the resultant weight assigned to $\boldsymbol{a}_{p_j}, j = 1, \cdots, N_1$, by the robust estimation cycle. The chromosome Ξ is then updated by using the following two-step self-adaptation operator.

Data Pruning. Reject the outliers detected in $G(\Xi)$ and keep the nonoutliers. It amounts to updating the chromosome bits $\xi_{p_j}, j = 1, \cdots, N_1$ as follows,

$$\xi_{p_j} = \leftarrow \begin{cases} 1 & \text{if} \quad w_{p_j} > 0 \\ 0 & \text{if} \quad w_{p_j} = 0 \end{cases} \quad (39)$$

Data Augmentation. Accept any consistent data points with respect to the current model $\hat{\boldsymbol{h}}$ that are originally in the irrelevant data set $\overline{G(\Xi)}$. This is accomplished as follows. Let the median of the absolute model residuals obtained in the robust estimation cycle be e_{med}. Compute the residual e_{q_k} for each $\boldsymbol{a}_{q_k} \in \overline{G(\Xi)}$ using (22) by substituting in the current model $\hat{\boldsymbol{h}}$. Then update the chromosome bits $\xi_{q_k}, k = 1, \cdots, N - N_1$ by,

$$\xi_{q_k} = \leftarrow \begin{cases} 1 & \text{if} \quad |e_{q_k}| \le e_{med} \\ 0 & \text{if} \quad |e_{qk}| > e_{med} \end{cases} \quad (40)$$

A self-adapted chromosome represents a new partition of the data set which supports the estimated motion model $\hat{\boldsymbol{h}}$. In essence, the self-adaptation operator utilizes the outlier rejection capability of the biweight robust estimator. Meanwhile, consistent data points that are originally excluded can now be included by the data augmentation procedure, ensuring that the extracted motion group is the largest possible.

B.3 Optic Flow Motion Fitness Function

The fitness function of a chromosome $\Xi = \xi_1 \xi_2 \cdots \xi_N$ is evaluated by two sources. Let $\boldsymbol{A}$ denote the design matrix composed by using the relevant data set $G(\Xi)$, i.e., $A = (\boldsymbol{a}_{p_1}^T, \cdots, \boldsymbol{a}_{p_{N_1}}^T)^T$. Let $\hat{\boldsymbol{h}}$ be the least-squares motion estimate by using $G(\Xi)$. The first fitness source is the normalized mean least-squares model fitting error $f_e(\Xi)$, i.e.,

$$f_e(\Xi) = \frac{\left\|\boldsymbol{A}\hat{\boldsymbol{h}}\right\|^2}{N_1 \left\|\hat{\boldsymbol{h}}\right\|^2}. \quad (41)$$

This value measures the "goodness" of the relevant data set $G(\Xi)$ with respect to the motion model $\hat{\boldsymbol{h}}$. In general, a more consistent motion group has smaller f_e value. Practically, data points in a coherent motion group are spatially close and share similar velocities. Thus, the second fitness source is the compactness measure of $G(\Xi)$. This is reflected as the generalized variance (or the determinant) of the sample covariance matrix $\boldsymbol{S}$ defined over $G(\Xi)$, i.e.,

$$f_v(\Xi) = |\boldsymbol{S}|, \quad (42)$$

where

$$\boldsymbol{S} = \frac{1}{N_1 - 1} \sum_{j=1}^{N_1} (\boldsymbol{a}_{p_j} - \bar{\boldsymbol{a}})(\boldsymbol{a}_{p_j} - \bar{\boldsymbol{a}})^T$$
$$\bar{\boldsymbol{a}} = \frac{1}{N_1} \sum_{j=1}^{N_1} \boldsymbol{a}_{p_j}.$$

Since the two fitness measures may have different numerical orders, rank statistics [22] are used to define the overall fitness function as follows,

$$f(\Xi_i) = \left(\frac{C - rank(f_e(\Xi_i))}{C - 1}\right)^{\phi} + \left(\frac{C - rank(f_v(\Xi_i))}{C - 1}\right)^{\phi} \quad (43)$$

where $rank(f_e(\Xi_i))$ and $rank(f_v(\Xi_i))$ are the ranks of the model fitting error and compactness measure of Ξ_i, respectively, in ascending order for a chromosome pool of size C, and ϕ is a sharpening constant.

B.4 Chromosome Pool Convergence

By repetition of the four genetic operators, the chromosome pool will converge to certain patterns [9]. In case of one motion group, the chromosome pool will hopefully become nearly uniform with each chromosome indicating the same motion group. In case of multiple motion groups, however, there are two possible stable patterns. First, the chromosome pool may still be uniform since each chromosome is attracted to the same motion group. Then we can apply the same algorithm recursively on the remaining data portion to extract other motions. Second, the chromosome pool can be piecewise uniform in the sense that there exists several repeated chromosome patterns, each indicating one distinct motion group. In other words, multiple consistent motion groups are concurrently detected in a single application of the algorithm. The occurrence of these two situations depends on the respective noise structures and data compactness among the participating motion groups. If the motion groups have similar noise structures and data compactness, their fitness values will be close in magnitude so that they are equally capable of surviving in the chromosome pool. Otherwise, the motion group with smaller noises and more compact data distribution will first dominate the chromosome pool.

If the chromosome pool fails to converge to a uniform or piece-wise uniform pattern in a predefined number of generations, we consider that no more motion group is left out in the remaining data set.

B.5 Robust Genetic Partitioning Algorithm

The robust genetic partitioning algorithm is summarized as follows:

1) *Chromosome Pool Initialization.* A set of C chromosomes of length N are randomly generated if no other initial partitioning information is available.

2) *Reproduction.* Collect the relevant data subset $G(\Xi_i)$ for

each chromosome Ξ_i, compute its fitness value $f(\Xi_i)$ by (43) and its reproduction probability $p(\Xi_i)$ by (37). Then, reproduce Ξ_i with the probability $p(\Xi_i)$ and put in the mating pool.

3) *Crossover.* Randomly pair off the chromosomes in the mating pool into couples. With a small probability P_c, exchange chromosome pieces split at a randomly selected bit location for each couple (see the basic genetic algorithm).
4) *Mutation.* Reverse each bit of a chromosome with a very small probability P_m.
5) *Self-Adaptation.* Collect the relevant data set $G(\Xi_i)$ for each chromosome Ξ_i. Submit $G(\Xi_i)$ to the robust estimation cycle to obtain the estimated motion model $\hat{\boldsymbol{h}}$ and the associated weight matrix $\boldsymbol{W}$. Update the chromosome Ξ_i by the data pruning step (39) and data augmentation step (40).
6) Repeat steps 2) to 5) until the chromosome pool converges or fails to converge after a specified number of generations.

The self-adaptation operator plays a key role in this algorithm. By using the robust estimation method to reject irrelevant data, a consistent data subset can be extracted from a large variety of combinations of partially correct partitions. In contrast, the basic genetic algorithm depends only the pure bit manipulations of the chromosomes; thus the chances to compose a consistent data subset are rare indeed. In our experiments, the self-adaptive genetic algorithm usually converges in 5–15 generations. The basic genetic algorithm can hardly converge within a reasonable number of iterations.

B.6 Limitations

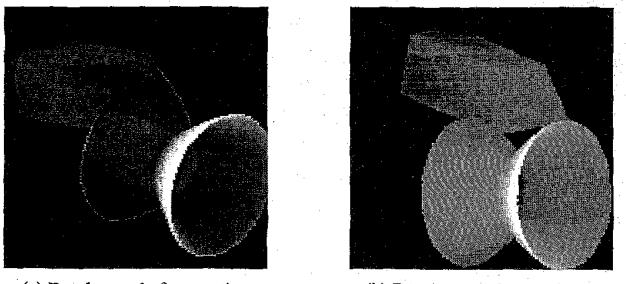
The proposed algorithm essentially searches through various combinations of the input data set. Thus, the computation cost increases exponentially with the data set size. A more vigorous analysis of its computational complexity is however complicated by the fact that the general convergence of the genetic algorithm is hard to quantify [9]. Thus, we indicate that from the computational perspective the proposed algorithm is more suitable for sparse optic flow field with less data points than dense optic flow field. To relieve this problem when dealing with a large data set (e.g., dense optic flow field), we may have two choices: 1) use a *divide-and-conquer* approach. That is, the data set can be divided into smaller subsets each of which is separately handled and the results are appropriated combined; and 2) subsample the dense optic flow field to reduce the data set size. Another related limitation is the number of multiple motion groups one can handle at a time. To ensure that each participating motion group is not under-represented in the data set, the number of motion groups handled simultaneously is not expected to be very large. This is related to the breakdown point of this robust algorithm. In the experiments, we find that the algorithm can handle up to four equally sized motion groups without difficulty. This corresponds to a breakdown point of about 80%. When handling more motion groups, we also suggest to use the *divide-and-conquer* approach in hoping that only a few motion groups are contained in each data subset. This is analogous to the block- or window-wise approaches in many image processing techniques.

VI. Experimental Results

We first present two illustrative examples using synthetic and real data sets. Then a large number of simulations are conducted to evaluate the overall performance of the proposed algorithm.

A. Synthetic 3-D Rigid Motion

Fig. 1a shows a synthetic 3-D scene consisting of two visually overlapping rigid objects—"drum and stake." The drum undergoes a rigid motion with instantaneous rotational velocity (0.0, 0.0, 1.7453) and translational velocity (–35.0, –100.0, 0.0); the stake with instantaneous rotational velocity (0.4363, 0.0, -1.3089) and translational velocity (25.0, 0.0, 0.0). Fig. 1b shows the new scene after 0.2 sec. The true optic flow field projected by the scene is shown in Fig. 2a where the data have been subsampled and scaled. Fig. 2b shows the contaminated optic flow field by Gaussian noises (SNR = 55) and 20% randomly altered outliers. The robust genetic partitioning algorithm is then applied. Fig. 2c shows the optic flow field segmentation where □ and o represent the two consistent motion groups, respectively, and • the detected outliers. Fig. 2d shows the recovered optic flow field. The estimated rotational and translation velocities for the drum are (–0.0008, –0.0003, 1.7435) and (–35.1054, –99.99603, –0.7338), respectively, and for the stake (0.4393, –0.0045, –1.3073) and (24.9984, 0.1112, 0.2573), respectively. It is seen that the multistructured and partially incorrect optic flow field is well partitioned and interpreted with high accuracy.



(a) Depth map before motion (b) Depth map after motion

Fig. 1. The "drum" and "stake" 3D scene undergoing rigid body motion. (a) The depth map before motion; (b) The depth map after 0.2 sec.

B. Outdoor Vehicle Motion

A motion sequence containing a moving army truck is retrieved from the image archive used in the 1991 IEEE Workshop on Visual Motion [18]. The left monocular image sequence contains 24 image frames taken by a fixed-focus A.M.I/Bronica SQ-AM metric 70 mm cameras with 40 mm nominal focal length. The world coordinate system is right-handed with its x and z axes on the ground plane and y axis vertically upward. The camera is mounted on the ground such that its optical axis roughly points to the negative z-direction of the world coordinate system. Several target points had been

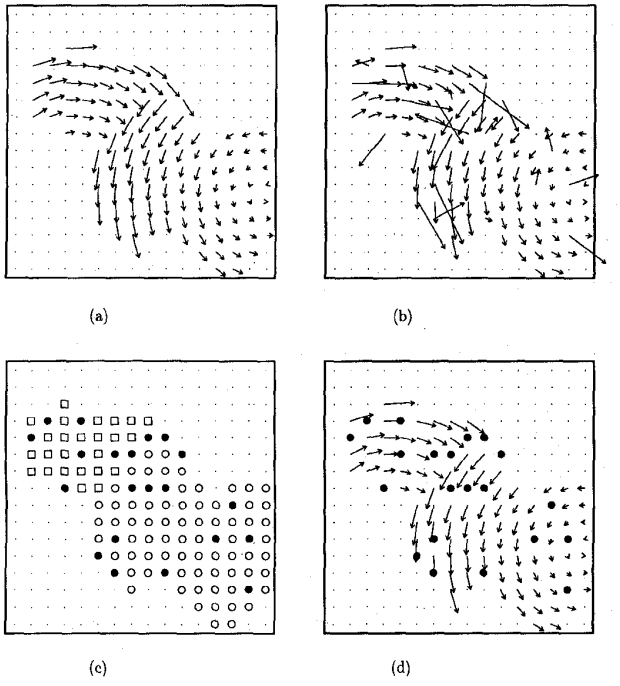


Fig. 2. (a) The true projective optic flow field; (b) The noisy and partially incorrect optic flow field; (c) The computed partition of the optic flow field where □ and ∘ represent two consistent motion groups and • represents the detected outliers; (d) The reconstructed optic flow field where • represents the detected outliers.

marked on the truck and the 2-D image coordinates of these points were used as input data for motion estimation [17]. Fig. 3a and 3b show the 384 × 512 subimages of 13th and 14th original image frames in the sequence, respectively. In order to fit this data set into our experiment, we first have to generate more target points on the vehicle. This is because that only four target points are visible in the original frames, which are sufficient for multiframe based motion estimation algorithm [17] but not for our two-view linear optic flow motion algorithm. As mentioned, the linear algorithm requires at least eight data points to uniquely determine a motion. Hence, 30 target points on the vehicle are selected and their matches are computed by using an adaptive matching algorithm [15] and further manually verified. Secondly, the original sequence concerns only with the vehicle motion. To create a multiple motion scenario, we treat the background motion (due to camera panning) as the second motion. Again 30 background target points are chosen and their matches found. All 60 target points are combinely marked as "+" in Fig. 3a and 3b. Following the procedures in [18], the target points are converted from pixel coordinates to image coordinates and further corrected by using camera calibration parameters. The mixture optic flow data set is thus composed by $\{(X_i, Y_i), (u_i, v_i)\}$, $i = 1, \cdots, 60$ where (X_i, Y_i) is the image coordinates of the ith target point in the 13th frame and (u_i, v_i) is the displacement of the ith pair of the target points.

The linear optic flow motion algorithm, the biweight robust estimator and the robust genetic partitioning algorithm are sequentially tested. We emphasize that the former two algorithms are supplied with the correct data subset for each motion; and for the last algorithm, the mixed data set is used as input data. Table I summaries the estimated motion parameters for the three algorithms where the rotational velocity $(\hat{\omega}_1, \hat{\omega}_2, \hat{\omega}_3)$ is converted to the equivalent representation of a rotation axis plus the rotation angle around the axis.

The established "ground truth" for the vehicle motion has not been revealed [18]. Furthermore, it is reported that the existing two-view point correspondence and three-view line correspondence algorithms all produce unreliable rotation estimates [17] since the angle subtended by the truck is too small (about 5%). In addition, the target points used by different algorithms are different. Therefore, we do not attempt to compare our motion estimates with those obtained by long sequence algorithms [17]. Roughly speaking, in the camera-centered coordinate system, the background moves to the right, inducing a major positive x translational component while the truck moves to the left, inducing a major negative x translational component. Since the camera also moves downwards, we expect the rotation axis to be roughly around the positive x axis for both motions. The linear algorithm produces the motion parameters that are basically in keeping with these observations. The robust algorithm detects a number of outliers for each motion group. Most importantly, the genetic partitioning algorithm is able to separate the mixture data set into two coherent motion groups and a small number of outliers; the estimated motion parameters are consistent with the linear algorithm and robust algorithm. Fig. 3c shows the partition of the optic flow field where □ and Δ denote the two motion groups, respectively, and "+" denote the detected outliers. We point out that the outliers can be further classified into one of the motion groups when necessary.

C. Simulation Protocol

Our simulations use the following experimental protocol to generate experimental data:

Object Points. The 3-D object surface points are randomly selected within a cuboid, i.e., $\boldsymbol{p}_i = (x_i, y_i, z_i)^T \in [10, 30] \times [10, 30] \times [30, 60]$, $i = 1, \cdots, N$.

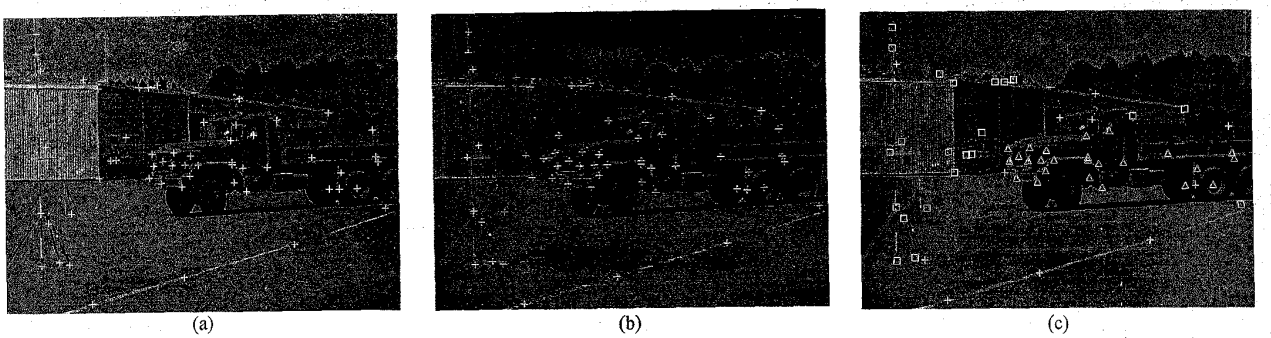
Motion Group. The instantaneous rotational angular velocity $(\omega_1, \omega_2, \omega_3)$ and the translational velocity (k_1, k_2, k_3) are randomly selected, i.e., $\omega_i \in [0.5, 5.5]$ and $k_i \in [1.0, 20.0]$ for $i = 1, 2, 3$.

Optic Flow Image Points. The 3-D points $\boldsymbol{p}_i$ are projected to the image plane $z = 1$ to obtain the 2-D image points (X_i, Y_i) and the ideal velocities (u_i, v_i) by (2) and (3).

Noise Process. The optic flow image points are contaminated by Gaussian noises or outliers. That is, independent identically distributed Gaussian noises are added to a portion of optic flow velocities (u_i, v_i) and the remaining optic flow velocities are randomly altered to form outliers.

Control Parameters. There are several parameters to be controlled to simulate various situations. These parameters are the data sample size N, the outlier proportion ϵ, and the Gaussian signal-to-noise ratio (SNR) defined as

$$SNR = 20 \log\{\textstyle\sum_i \sqrt{u_i^2 + v_i^2} \,/\, (N(1-\epsilon)\sigma)\}$$



(a) (b) (c)

Fig. 3. Outdoor vehicle and background motions (a) The 13th image frame with marked target points; (b) The 14th image frame with the corresponding target points; (c) The optic flow field partition by the robust genetic partitioning algorithm where □ and Δ denote the two distinct motion groups, respectively, and "+" denote the detected outliers.

TABLE I
ESTIMATED MOTION PARAMETERS FOR THE OUTDOOR VEHICLE AND BACKGROUND MOTIONS

algorithm	motion group	rotation axis n_1	n_2	n_3	angle (degree)	translation x	y	z	#(outliers)
linear	truck	0.9592	0.0816	-0.2708	1.3759	-0.9898	-0.0021	-0.1196	N.A.
	background	0.9390	-0.3008	0.1668	0.5109	0.9422	0.3257	0.0772	N.A.
robust	truck	0.9589	0.0840	-0.2708	1.4232	-0.9849	-0.0077	-0.1522	4
	background	0.9299	-0.3568	0.0891	0.5693	0.9239	0.3750	0.0744	3
robust genetic partition	truck	0.9588	0.0518	-0.2791	1.3972	-0.9873	-0.0165	-0.1276	11
	background	0.9286	-0.3693	0.0372	0.4596	0.9311	0.3575	0.0361	

where the summation is taken over nonoutliers and σ is the noise standard deviation.

Result Averaging. For each combination of the control parameters, 100 trials are tested and the averaged results are reported.

D. Baseline Model Evaluation

In this set of simulations, the behavior of the linear optic flow motion algorithm is characterized for recovering single 3-D rigid motion under Gaussian noises. Fig. 4 shows the relative accuracy of the estimated motion parameters with respective to the true values under various conditions. As expected, the accuracy decreases as the SNR decreases, or the sample size decreases.

E. Single Motion Group

In this set of experiments, we test the proposed robust genetic partitioning algorithm for recovering single 3-D rigid body motion from the motion data sets composed of one consistent motion group and 10% to 70% outliers. The size of the chromosome pool is fixed as 50 in all trials. For each trial, the algorithm usually converges within five generations and takes less than 30 sec in average on SunSparc LX Workstations. Fig. 5 shows the estimation accuracy under various control parameters. As seen, the algorithm effectively suppresses the influence of outliers and registers only slightly increased estimation errors as compared to the baseline model performance. The segmentation errors are summarized in Table II where the average number of incorrectly partitioned data is reported. In the table, "r1" refers to the number of outliers that are incorrectly classified as motion data, while "r2" refers to the number of motion data that are incorrectly classified as outliers. Note that outliers being classified as motion data can greatly affect the estimation accuracy. This is partially reflected in Fig. 5 for the case $\epsilon = 70\%$ which has much reduced accuracy because the corresponding "r1" value is high. On the other hand, motion data being classified as outliers has unnoticeable impact on the estimation.

F. Multiple Motion Groups

In this set of experiments, we test the proposed algorithm for recovering two or three 3-D rigid motions from multi-structured and partially incorrect motion data. The motion groups are independently generated and mixed as a single motion data set. For the two-motion case, each motion group occupied 45% data and the remaining 10% data are outliers. For the three-motion case, each motion group occupied 30% data and the remaining 10% data are outliers.

When applying the algorithm, we find that the two or three motion groups are mostly sequentially extracted. This reflects the different noise structures and data compactness among these motion groups which may be due to the random data generation mechanism. In order to characterize the different

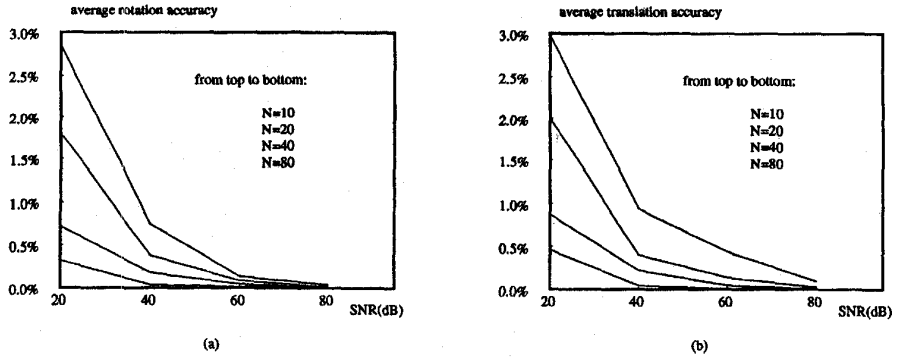


Fig. 4. Baseline model evaluation with the motion data contaminated only by Gaussian noises (a) Estimation accuracy of the instantaneous rotational angular velocity; (b) Estimation accuracy of the instantaneous translational velocity.

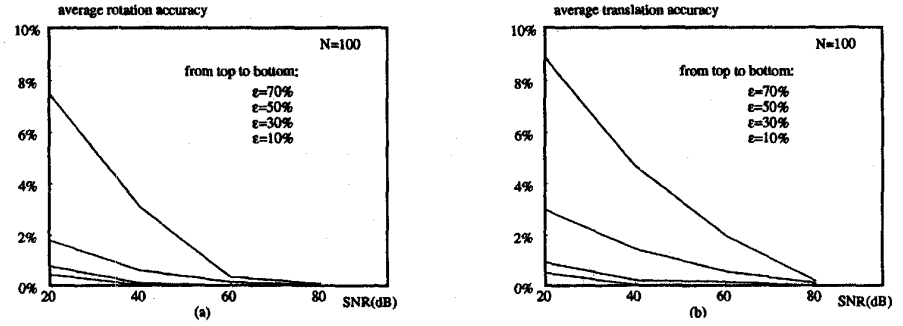


Fig. 5. Single-motion case where the motion data are contaminated by both Gaussian noises and outliers (a) Estimation accuracy of the instantaneous rotational angular velocity; (b) Estimation accuracy of the instantaneous translational velocity.

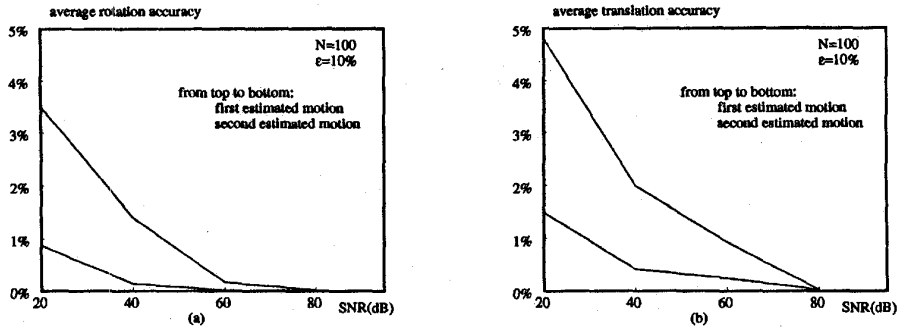


Fig. 6. Two-motion case where the motion data contain two mingled motion groups as well as outliers (a) Estimation accuracy of the instantaneous rotational angular velocity; (b) Estimation accuracy of the instantaneous translational velocity.

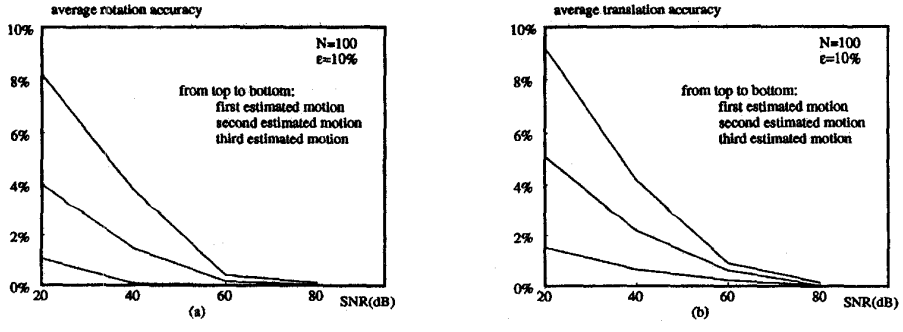


Fig. 7. Three-motion case where the motion data contain three mingled motion groups as well as outliers (a) Estimation accuracy of the instantaneous rotational angular velocity; (b) Estimation accuracy of the instantaneous translational velocity.

TABLE II
AVERAGE NUMBER OF WRONGLY SEGMENTED MOTION DATA POINTS FOR SINGLE MOTION SEGMENTATION AND ESTIMATION (N = 100)

SNR	ε = 10%		ε = 30%		ε = 50%		ε = 70%	
	r1	r2	r1	r2	r1	r2	r1	r2
80	0.00	0.00	0.00	0.75	0.01	0.92	0.03	1.58
60	0.00	0.00	0.01	1.04	0.11	1.32	0.28	2.35
40	0.00	0.00	0.03	1.88	0.14	2.15	0.42	2.89
20	0.00	0.00	0.08	2.55	0.27	3.27	0.74	4.18

r1: outliers classified as motion points
r2: motion points classified as outliers

TABLE III
AVERAGE NUMBER OF WRONGLY SEGMENTED MOTION DATA POINTS FOR TWO-MOTION SEGMENTATION AND ESTIMATION (N = 100)

SNR	first motion		second motion	
	r1	r2	r1	r2
80	0.02	1.44	0.00	0.00
60	0.26	1.85	0.00	0.00
40	0.37	2.57	0.00	0.00
20	0.62	3.08	0.00	0.00

r1: outliers classified as motion points
r2: motion points classified as outliers

TABLE IV
AVERAGE NUMBER OF WRONGLY SEGMENTED MOTION DATA POINTS FOR THREE-MOTION SEGMENTATION AND ESTIMATION (N = 100)

SNR	first motion		second motion		third motion	
	r1	r2	r1	r2	r1	r2
80	0.03	1.45	0.01	1.53	0.00	0.66
60	0.30	2.48	0.11	1.89	0.00	1.05
40	0.46	3.05	0.21	2.49	0.00	1.76
20	0.83	4.31	0.36	3.18	0.00	2.34

r1: outliers classified as motion points
r2: motion points classified as outliers

behaviors of the algorithm in sequentially extracting motion groups, we provide its separate performance statistics at different stages. Fig. 6 and Fig. 7 show the estimation accuracy for the two-motion case and three-motion case, respectively, where the "first estimated motion" refers to the average performance of extracting the first motion group by the algorithm; "second estimated motion" refers to the average performance of extracting the second motion group by the algorithm; and so forth. It is found that the sequentially extracted motion groups have increasingly higher accuracy due to the removal of the preceding motion group(s). In these trials, the algorithm converges within 5–15 generations for extracting each motion group, and the completion of each trial takes 30–120 sec on SunSparc LX Workstations. The segmentation errors are summarized in Table III and Table IV, which are consistent with the estimation accuracy. These results reveal that the proposed algorithm is capable of recovering multiple motion groups with high accuracy under partially correct conditions.

VII. CONCLUSION

In the paper, we have proposed a robust genetic algorithm to handle multiple optic flow motions under various adversary conditions. The proposed algorithm is generally applicable to other vision problems that can be formulated as general regression problems. Combining robust estimators with genetic algorithms is a promising line of research and a novel idea. It offers a new framework of determining multistructured data sets, as well as partially incorrect data sets. These kinds of data sets are the most realistic ones in many vision problems. We intend to investigate the convergence properties of the modified genetic algorithm in the near future..

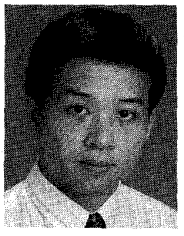
ACKNOWLEDGMENTS

The authors want to thank the reviewers (especially reviewer 1), for the detailed revision comments which lead to the improvement of this paper. This work was supported by the National Science Foundation under Grant No. IRI 9112649 and by the National Aeronautics and Space Administration Applied Information Systems Research program award NRA-93-OSSA-09 and NAG-2573 managed by Glenn Mucklow.

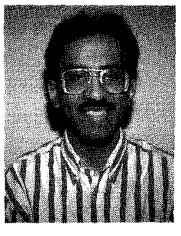
REFERENCES

[1] G. Adiv, "Determining three-dimensional motion structure from optic flow generated by several moving object," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7, no. 4, pp. 384-401, 1985.
[2] G. Adiv, "Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 477-489, 1989.
[3] J.K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of image—a review," *Proc. IEEE*, vol. 76, no. 8, pp. 917-935, 1988.
[4] A.R. Bruss and B.K. Horn, "Passive navigation," *Computer Vision Graphics Image Processing*, vol. 21, pp. 3-20, 1983.
[5] H.H. Chen and T.S. Huang, "Matching 3-D line segments with applications to multiple-object motion estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 1,002-1,008, 1990.

[6] T. Darrell and A. Pentland, "Robust estimation of a multi-layered motion representation," *Proc. IEEE Workshop Visual Motion*, Princeton, N.J., pp. 173-178, 1991.
[7] J.N. Driessen, L. Boroczky, and J. Biemond, "Pel-recursive motion field estimation from image sequences," *J. Visual Comm. and Image Representation*, vol. 2, no. 3, pp. 259-280, 1991.
[8] W.A. Fuller, *Measurement Error Models*, New York: John Wiley & Sons, 1987.
[9] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Mass.: Addison-Wesley, 1989.
[10] R.M. Haralick, H. Joo, C.N. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim, "Pose estimation from corresponding point data," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 19, pp. 1,426-1,446, Aug. 1989.
[11] F. Heitz and P. Bouthemy, "Multimodal estimation of discontinuous optical flow using Markov random fields," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 12, pp. 1,217-1,237, 1993.
[12] J.H. Holland, "Adaptation in natural and artificial systems," *Proc. IEEE*, Ann Arbor, Mich.: Univ. of Michigan Press, 1975.
[13] B.K.P. Horn and E.J. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 23, pp. 185-203, 1981.
[14] T.S. Huang and S.D. Blostein, "Robust algorithms for motion estimation based on two sequential stereo image pairs," *IEEE Conf. Computer Vision Pattern Recognition*, pp. 518-523, 1985.
[15] Y. Huang and X. Zhuang, "An adaptively refined block matching algorithm for video coding," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 5, no. 1, 1995.
[16] P.J. Huber, *Robust Statistics*. New York: John Wiley & Sons, 1981.
[17] Y. Liu and and T.S. Huang, "Vehicle-type motion estimation from multi-frame Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 8, pp. 802-808, 1993.
[18] Y. Liu and and T.S. Huang, "A sequence of stereo image data of a moving vehicle in an outdoor scene," Tech. Report, Beckman Institute, Univ. of Illinois at Urbana-Champaign, 1990.
[19] H.C. Longuet-Higgins, "A computer program for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133-135, 1981.
[20] P. Meer, D. Mintz, and A. Rosenfeld, "Robust regression methods for computer vision: A review," *Int'l J. Comp. Vision*, vol. 6, pp. 59-70, 1991.
[21] D.W. Murray and B.F. Buxton, "Scene segmentation from visual motion using global optimization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 2, pp. 220-228, 1987.
[22] R.H. Myers, *Classical and Modern Regression with Applications*. Boston, Mass.: PWS-KENT Publishing Company, 1990.
[23] H.-H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector field from image sequences," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 5, pp. 565-593, 1986.
[24] J.W. Roach and J.K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 6, pp. 554-562, 1980.
[25] M.A. Snyder, "On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1,105-1,114, 1991.
[26] W.B. Thompson, "Combining motion and contrast for segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, pp. 543-549, 1980.
[27] R.Y. Tsai and T.S. Huang, "Uniqueness and estimation of 3-D motion parameters of rigid objects with curved surfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 13-26, 1984.
[28] G. Tziritas, "Recursive and/or iterative estimation of the two-dimensional velocity field and reconstruction of three-dimensional motion," *Signal Processing*, vol. 16, pp. 53-72, 1989.
[29] Ullman, *The Interpretation of Visual Motion*. Cambridge, Mass.: MIT Press, 1979.
[30] A. Verri and T. Poggio, "Against quantitative optical flow," *Proc. First Int'l Conf. Computer Vision*, London, England, pp. 171-180, 1987.
[31] J.Y.A. Wang and E.H. Anderson, "Representing moving images with layers," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 625-638, 1994.
[32] J. Weng, N. Ahuja, and T.S. Huang, "Two-view matching," *Proc. Second Int'l Conf. Computer Vision*, Tampa, Fla., pp. 64-73, 1989.
[33] J. Weng, N. Ahuja, and T.S. Huang, "Optimal motion and structure estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 864-884, 1992.
[34] B.L. Yen and T.S. Huang, "Determining 3-D motion and structure of a rigid body using the spherical projection," *Computer Vision Graphics Image Processing*, vol. 21, pp. 21-32, 1983.
[35] G.J. Young and R. Chellappa, "Statistical analysis of inherent ambiguities in recovering 3-D motion and structure from a noisy flow field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 995-1,013, 1992.
[36] Z. Zhang and O.D. Faugeras, "Estimation of displacements from two 3-D frames obtained from stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 12, pp. 1,141-1,156, 1992.
[37] X. Zhuang, T.S. Huang, and R.M. Haralick, "Two-view motion analysis: A unified algorithm," *J. Optical Soc. Am.*, vol. 3, no. 9, pp. 1,492-1,500, ser. A, 1986.
[38] X. Zhuang, T.S. Huang, and R.M. Haralick, "A simplified linear optic flow-motion algorithm," *Computer Vision, Graphics, and Image Processing*, vol. 42, pp. 334-344, 1988.
[39] X. Zhuang, T. Wang, and P. Zhang," A highly robust estimator through partially likelihood function modeling and its application in computer vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 19-35, 1992.
[40] X. Zhuang and Y. Huang, "Robust 3D-3D pose estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 818-824, 1994.
[41] X. Zhuang and R.M. Haralick, "Developing robust techniques for computer vision," *Proc. Int'l Workshop Robust Computer Vision*, Seattle, Wash., Oct. 1990.



Yan Huang received the BE degree in computer science and engineering from Zhejiang University, China, in 1990 and the MS and PhD degree (both in electrical engineering) from the University of Missouri, Columbia, in 1992 and 1995, respectively. He is currently at AT&T Bell Laboratories in New Jersey. Dr. Huang received a superior graduate achievement award from the University of Missouri in 1995. He worked for NeoPath, Inc., Amiable Technologies, Inc., and NASA Goddard Space Flight Center, during the summers of 1992, 1993, and 1994, respectively. He has over 20 technical publications including 10 journal publications. His research interests include video and image compression, computer vision, image processing, pattern recognition, and neural networks.

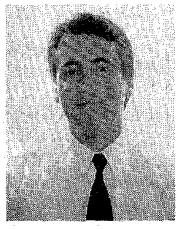


K. Palaniappan (S'84–M'90) received the BASc and MASc degrees in systems design engineering from the University of Waterloo, UIUC, Waterloo, Canada and the PhD degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1991. As an undergraduate student he had work terms with various companies including Ontario Hydro, Canadian Ministry of Environment, Bell Canada, and Bell Northern Research. In 1981 he was selected as an IAESTE exchange student with Preussen Elektra in Landesbergen, Germany. He has been a teaching and research assistant in the Department of Systems Design Engineering at the University of Waterloo, in the Coordinated Science Laboratory at UIUC, in the Department of Electrical and Computer Engineering at UIUC and in the Department of Microbiology at UIUC. In the summer of 1986 he was at Bell Northern Research in Montreal, Canada working on document image analysis. Since 1991 he has been working at NASA Goddard Space Flight Center (GSFC) (affiliated with USRA as a senior research scientist) in the Mesoscale Atmospheric Processes Branch. His research interests include scientific visualization, image analysis, computer vision, parallel algorithms, pattern recognition and visual databases. He has been working on the development of a

novel tool known as the Interactive Image Spread Sheet (IISS) for the analysis of extremely large remote sensing datasets and numerical weather model data, and on satellite-based stereo and non-rigid motion analysis for automatic cloud tracking using parallel algorithms.

Dr. Palaniappan received a NASA Outstanding Achievement Award (Mesoscale Dynamics and Precipitation Branch) in 1993. He received Universities Space Research Association's Creativity and Innovation Science Award in 1993, Natural Sciences and Engineering Research Council of Canada Scholarship from 1982–1988, University of Waterloo Engineering Scholarship from 1982–1984, and an Ontario Scholar Award in 1978.

Xinhua Zhuang (SM'92) is currently a professor of electrical and computer engineering at the University of Missouri at Columbia. Professor Zhuang has over 150 publications in the areas of signal processing, speech recognition, image processing, machine vision, pattern recognition, and neural networks. He has made contributions to six books. He has received a number of awards including a NATO Advisory Group of Aerospace Research and Development (AGARD) fellowship; Natural Science Foundation grants of China; K.C. Wong Education Foundation grants of Hong Kong; National Science Foundation grants; and NASA HPCC grants. He has held a number of consulting positions including Siemens, Panasonic, NeoPath Inc., and NASA. Professor Zhuang has been affiliated with a number of schools and research institutes prior to joining the University of Missouri in 1990, including the Zhejiang University of China, the University of Washington, the University of Illinois, the University of Michigan, the Virginia Polytechnic Institute and State University, and the Research Institute of Computer Technology of China. Professor Zhuang joined the IEEE in 1992 as a senior member and presently serves as associate editor of *IEEE Transactions on Image Processing*.



Joseph E. Cavanaugh received his bachelor's degree in computer science and pure mathematics in 1986 from the Montana College of Mineral Science and Technology. He received his master's degree in statistics in 1988 from Montana State University, and his PhD in statistics in 1993 from the University of California at Davis. Dr. Cavanaugh joined the faculty of the University of Missouri at Columbia in 1993 and is currently an associate professor in the Department of Statistics. His research interests include model selection, time series analysis, linear models, state-space modeling, Bayesian analysis, and simulation and resampling methodologies.