Estimating 3-D Egomotion from Perspective Image Sequences

Wilhelm Burger and Bir Bhanu, *Senior Member, IEEE*

Abstract— **This paper deals with the computation of sensor motion from sets of displacement vectors obtained from consecutive pairs of images. The problem is investigated with emphasis on its application to autonomous robots and land vehicles. First, the effects of 3-D camera rotation and translation upon the observed image are discussed and in particular the concept of the focus of expansion (FOE). It is shown that locating the FOE precisely is difficult when displacement vectors are corrupted by noise and errors. A more robust performance can be achieved by computing a 2-D region of possible FOE-locations (termed the fuzzy FOE) instead of looking for a single-point FOE. The shape of this FOE-region is an explicit indicator for the accuracy of the result. It has been shown elsewhere that given the fuzzy FOE, a number of powerful inferences about the 3-D scene structure and motion become possible. This paper concentrates on the aspects of computing the fuzzy FOE and shows the performance of a particular algorithm on real motion sequences taken from a moving autonomous land vehicle.**

Index Terms—**Autonomous mobile robot, dynamic scene analysis, fuzzy focus of expansion, motion analysis, passive navigation, sensor motion estimation.**

I. Introduction

THE problem of determining the motion parameters of a moving camera relative to its environment from a sequence of images is important for the application of computer vision in mobile robots. Short-term control, such as steering and braking, path stabilization, navigation, and obstacle avoidance are all tasks that can effectively utilize this information [11], [12]. Several researchers have addressed this problem directly [3], [10], [17], [19] or indirectly by determining the motion parameters of a single rigid object with respect to a stationary camera [9], [14], [21], [28], [30]. Since the (stationary) environment can be considered as one large rigid object, these two approaches are equivalent. A prerequisite for any existing method is to estimate the 2-D motion that occurs in the image between consecutive frames. Two basic methods have been proposed for this purpose, which are quite distinct.

The *gradient method* [20], [29] uses spatial and temporal gray-level variations to estimate the instantaneous velocity or *optical flow* at each location in the image. It relies on sufficient object texture, continuous motion, and small displacements between subsequent frames. Since the magnitude of flow can only be determined in the direction of 2-D gradient, i.e., perpendicular to the tangent of a boundary, the flow vectors cannot be computed locally. This is commonly referred to as the *aperture problem* [31]. Global smoothing of the flow field has been proposed, which gives rise to problems at flow discontinuities, such as object boundaries. This seems to be almost a paradox, since intuitively motion estimates should be obtained most easily at exactly those locations.

The *displacement method* [2], [4] uses the parts of the image where discontinuities in brightness or motion occur (which are a source of problems in the gradient method). Significant features such as line segments, distinct dark or bright spots, or corners in two consecutive frames are selected and matched, rendering a field of *displacement vectors* for those features. This results in a formulation of 2-D motion as discrete displacements as compared to the velocity-based formulation which characterizes the gradient method. Ideally, the selected features should not only be matched between subsequent frames, but tracked over multiple frame sequences. Two problems arise during this process. The *first* problem is the selection and reliable location of significant features in consecutive frames, especially when the images are noisy. Individual features are commonly extracted by applying local window operations, like Moravec's "interest operator" [23] as a classic example. The application of this approach on a mobile land-based robot faces additional difficulties. In this particular scenario, a forward-looking camera moves approximately parallel and at relatively small distance to the ground. Therefore, visual objects in the environment "approach" the camera from almost infinite distance to as close as a few meters, before they leave the field of view. A point which was clearly distinguishable when observed from some distance, may lose its sharp outlines when the camera gets closer to it. This suggests some form of *range-dependent* feature-extraction from the image [18]. The *second* problem is finding reliable matches between the set of interesting points extracted from one frame and the set of interesting points extracted from the subsequent frame. This has been termed the *correspondence problem* [4] which is, although a nontrivial task in itself, assumed to be solved in the context of this work.

In spite of the persisting difficulties, the displacement method appears to be more promising than the gradient method. Not only is the information contained in discrete displacements fields of more practical use, but the problem of reliably computing optical flow is as hard as extracting and matching distinct features and has theoretical limitations [32]. In this work, we use a discrete feature matching approach, a local technique, where significant (i.e., "interesting" [23]) image events are localized in both images and subsequently matched upon similarity [4], [18]. This results in a field of displacement vectors between corresponding feature points.

From a given set of image displacement vectors, the 3-D structure of the scene and its relative motion towards the camera can be obtained simultaneously by solving a system of linear or nonlinear equations [14], [21], [28], [30]. While this technique is known to be numerically unstable under noisy conditions, recent work [15], [33] demonstrates that improvements are possible or that error estimates can be obtained.

Manuscript received August 20, 1987; revised May 16, 1990. Recommended for acceptance by W. B. Thompson. This work was supported by the Defense Advanced Research Projects Agency under Contract DACA 76-86-C-0017 and monitored by the U.S. Army Engineer Topographic Laboratories.

W. Burger was with Honeywell Systems and Research Center, 3660 Technology Drive, Minneapolis, MN 55418. He is now with the Institute of Systems Science, Johannes Kepler University, Linz, Austria.

B. Bhanu is with Honeywell Systems and Research Center, 3660 Technology Drive, Minneapolis, MN 55418.

IEEE Log Number 9038388.

0162-8828/90/1100-1040$01.00 © 1990 IEEE

When the camera is looking forward and the platform motion is characterized by a significant translation component (which is typical for most vehicles), another technique becomes attractive. It relies upon the fact that, under pure camera translation, all image features seem to diverge from a particular image location, called the *focus of expansion* (FOE), which marks the direction of vehicle heading [8], [20], [22], [24], [26]. This is of course only true for the stationary part of the scene but moving objects may be handled individually [1]. The advantage of this method is that the FOE, and consequently all parameters of the egomotion, can be computed entirely in terms of 2-D image coordinates regardless of the spatial structure of the scene.

We have adopted the FOE method as the basis of our approach. Unfortunately, computation of a *single* location for the FOE turns out to be a hard problem, mainly due to digitization errors, unreliable displacement vectors, and image noise. There seems to be psychological evidence [27] that humans also have difficulties in estimating the precise direction of heading in comparable situations. Our solution to the problem is not to search for a single FOE-location, but to obtain a two-dimensional FOE-region, which we call the "fuzzy FOE" [13]. Despite the apparent loss in accuracy, the Fuzzy FOE can be employed as a practical tool in dynamic scene analysis [6], [12]. In the following sections, we discuss the effects of camera motion upon the image and ways to decompose these effects. We distinguish between the *FOE-from-rotation* and *rotation-from-FOE* approaches, which are characterized by opposite search strategies in the multi-dimensional parameter space. Although the latter approach turns out to be superior, the actual location of the FOE can generally not be determined precisely under noisy conditions. A description of the fuzzy FOE algorithm is followed by some typical results on real image sequences which were taken from the autonomous land vehicle (ALV).

II. Image Effects of 3-D Camera Motion

It is well-known that any rigid motion of an object in space between two points in time can be decomposed into a combination of translation and rotation. While many researchers have used a velocity-based formulation of the problem [1], [26], the following treatment views motion in discrete time steps. Given the world coordinate system shown in Fig. 1, a translation $\boldsymbol{T} = (U\ V\ W)^T$ applied to a 3-D point $\boldsymbol{X} = (X\ Y\ Z)^T$ is accomplished through vector addition: $\boldsymbol{X}' = \boldsymbol{T} + \boldsymbol{X}$.

A 3-D rotation $\boldsymbol{R}$ about an arbitrary axis through the origin of the coordinate system can be described by successive rotations $\boldsymbol{R}_\phi$, $\boldsymbol{R}_\theta$, and $\boldsymbol{R}_\psi$ about its X-, Y-, and Z-axes, respectively. Thus, $\boldsymbol{X}' = \boldsymbol{R}\boldsymbol{X} = \boldsymbol{R}_\phi\boldsymbol{R}_\theta\boldsymbol{R}_\psi\boldsymbol{X}$, where

$$\boldsymbol{R}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \quad \boldsymbol{R}_\theta = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix},$$

$$\boldsymbol{R}_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

The transformation $\boldsymbol{M}$ for arbitrary rigid motion in three-space is thus given by

$$\boldsymbol{M} : \boldsymbol{X} \rightarrow \boldsymbol{X}' = \boldsymbol{R}_\phi\boldsymbol{R}_\theta\boldsymbol{R}_\psi(\boldsymbol{T} + \boldsymbol{X}), \tag{2}$$

with six degrees of freedom (ϕ, θ, ψ, U, V, and W). This decomposition is not unique because the translation could be applied after the rotation instead. Also, since the multiplication of the rotation matrices is not commutative, a different order of rotations would result in different amounts of rotation for each axis. For a fixed order of application, however, this motion decomposition is unique.

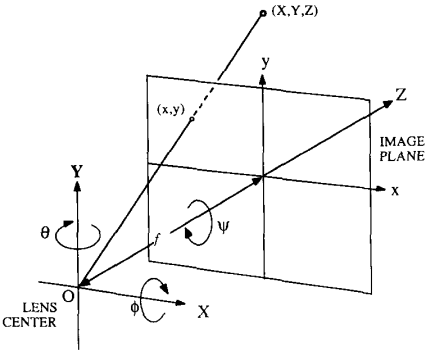

Fig. 1. Camera model showing the coordinate system, lens center, image plane, and angles of rotation. The origin of the coordinate system $\boldsymbol{O}$ is located at the lens center. The focal length f is the distance between the lens center and the image plane.

To model the environment of the vehicle, the camera is considered as being stationary and the environment as being moving as one single rigid object relative to the camera. The origin of the coordinate system (Fig. 1) is located in the lens center of the camera. The given task is to reconstruct the vehicle's egomotion from visual information. It is therefore necessary to know the effects of different kinds of vehicle motion upon the observed image.

Under perspective imaging, a point $\boldsymbol{X} = (X\ Y\ Z)^T$ in 3-D space is projected onto a location in the image plane $\boldsymbol{x} = (x\ y)^T$, with

$$x = \frac{fX}{Z} \text{ and } y = \frac{fY}{Z}, \tag{3}$$

where f is the focal length of the camera (see Fig. 1).

A. Effects of Pure Camera Rotation

The effects caused by pure camera rotation about an axis passing through the lens center are intuitively clear (see Fig. 1). For example, if the camera is rotated about the Z-axis (i.e., the optical axis), points in the image move along circles centered at the image location $\boldsymbol{x}_c = (0\ 0)$. In practice, however, we can assume that the amount of rotation about the Z-axis is relatively small and we therefore, only consider the more significant rotations about the X- and Y-axes.

Rotating the vehicle about the X-axis by an angle $-\phi$ and about the Y-axis by an angle $-\theta$ moves each 3-D point $\boldsymbol{X}$ to a new location $\boldsymbol{X}'$, given as

$$\boldsymbol{X} \rightarrow \boldsymbol{X}' = \boldsymbol{R}_\phi \cdot \boldsymbol{R}_\theta \cdot \boldsymbol{X}$$
$$= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ \sin\phi\sin\theta & \cos\phi & -\sin\phi\cos\theta \\ -\cos\phi\sin\theta & \sin\phi & \cos\phi\cos\theta \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{4}$$

Consequently $\boldsymbol{x}$, the image point of $\boldsymbol{X}$, moves to $\boldsymbol{x}'$ given by,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = f\frac{}{-X\cos\phi\sin\theta + Y\sin\phi + Z\cos\phi\cos\theta} \cdot \begin{bmatrix} X\cos\theta + Z\sin\theta \\ X\sin\phi\sin\theta + Y\cos\phi - Z\sin\phi\cos\theta \end{bmatrix}. \tag{5}$$

Inverting the perspective transformation from (3) and using it in (4) leads to the 2-D *rotation mapping* $r_\phi r_\theta$ which moves each image point $x = (x\,y)$ into the corresponding image point $x' = (x'y')$ under the sequence of 3-D rotations $R_\phi R_\theta$,

$$R_\phi R_\theta(X) : X \rightarrow X'$$

$$r_\phi r_\theta(x) : x = (x\,y) \rightarrow x' = (x'\,y')$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = f \frac{}{-x\cos\phi\sin\theta + y\sin\phi + f\cos\phi\cos\theta} \cdot \begin{bmatrix} x\cos\theta + f\sin\theta \\ x\sin\phi\sin\theta + y\cos\phi - f\sin\phi\cos\theta \end{bmatrix}. \quad (6)$$

It is important to notice that this transformation contains no 3-D variables, such that its effects can be simulated without knowing the distance of the observed points from the image plane. Therefore, the acquired image changes when the camera rotates around its lens center, but no additional, entirely new views of the environment are obtained. Ignoring the effects at the boundary of the image and errors due to the discretization of the image, we can assume that pure camera rotations merely map the image to itself.

To compute the amount of rotations from a pair of observations, we need to solve the inverse problem. Given are two image locations x_0 and x_1, which are the projections of a 3-D point X at time t_0 and time t_1, what are the amount of rotation ϕ and θ which when applied to the camera between instances t_0 and t_1, would move image point x_0 onto x_1, assuming that no camera translation has occurred?

If horizontal rotation R_ϕ and vertical rotation R_θ are applied to the camera separately, the points in the image move along hyperbolic paths [24]. If pure rotation R_θ were applied to the camera, a given image point $x_0 = (x_o y_o)$ would move on a path described by

$$r_\theta(x_0) : y^2 = y_0^2 \frac{f^2 + x^2}{f^2 + x_0^2}. \quad (7a)$$

Similarly, pure vertical rotation R_ϕ would move an image point $x_1 = (x_1 y_1)$ along a path described by

$$r_\phi(x_1) : x^2 = x_1^2 \frac{f^2 + y^2}{f^2 + y_1^2}. \quad (7b)$$

Since the composite 3-D rotation is modeled as two separate steps (R_θ followed by R_ϕ), the *rotation mapping*, $r_\phi r_\theta$, can also be separated into r_θ followed by r_ϕ. In the first step r_θ, rotation around the Y-axis, moves the original image point x_0 to an intermediate location x_c. Subsequently, r_ϕ would take point x_c to the final location x_1 by camera rotation around the X-axis. All this can be expressed as

$$r_{\phi\theta} = r_\phi r_\theta,$$

where

$$r_\theta : x_0 = (x_0\,y_0) \rightarrow x_c = (x_c\,y_c),$$

and

$$r_\phi : x_c = (x_c\,y_c) \rightarrow x_1 = (x_1\,y_1). \quad (8)$$

As shown in Fig. 2, the image point $x_c = (x_c\,y_c)$ is located at the intersection of a horizontal hyperbola passing through x_0 (7a) and a vertical hyperbola through x_1 (7b). The coordinates of the intersection point x_c are

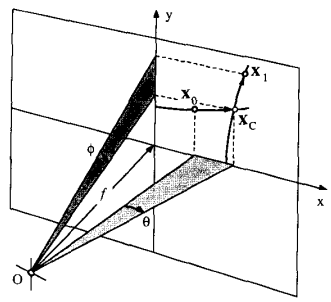

Fig. 2. Effects of camera rotation. Rotation about the Y-axis is applied first, which moves the arbitrary image point x_0 along a hyperbolic path to x_c. Subsequent rotation downwards about the X-axis moves x_c to x_1. The image location x_c is found by intersecting the two hyperbolae passing through x_0 and x_1.

$$x_c = f x_1 \left(\frac{f^2 + x_0^2 + y_0^2}{(f^2 + x_0^2)(f^2 + y_1^2) - x_1^2 y_0^2} \right)^{1/2} \quad (9a)$$

$$y_c = f y_0 \left(\frac{f^2 + x_1^2 + y_1^2}{(f^2 + x_0^2)(f^2 + y_1^2) - x_1^2 y_0^2} \right)^{1/2}. \quad (9b)$$

From this, the angles of rotation θ and ϕ (for this particular order of rotations) are finally obtained as (see Fig. 2),

$$\theta = \tan^{-1}\frac{x_c}{f} - \tan^{-1}\frac{x_0}{f}, \qquad \phi = \tan^{-1}\frac{y_c}{f} - \tan^{-1}\frac{y_1}{f}. \quad (10)$$

B. Effects of Pure Camera Translation

When the vehicle undergoes pure translation $T = (U\,V\,W)^T$ between time t_0 and t_1, every point X_i in the environment moves relative to the vehicle by the vector $-T$. Since every stationary point is affected by the same translation vector, these points actually move along imaginary parallel lines in 3-D space. It is a fundamental result from perspective geometry [16] that the images of parallel lines pass through a single point in the image plane called a *vanishing point* (Fig. 3). Therefore, when the camera moves forward along a straight line in space, image points seem to diverge from this vanishing point (the *focus of expansion*—FOE) or converge towards it (the *focus of contraction*—FOC) when the camera moves backwards.

Fig. 3 demonstrates that the straight line passing through the lens center O of the camera and the FOE is also parallel to the 3-D motion vectors of the environmental points. Therefore, the 3D vector $\vec{OF}$ (from lens center to FOE) points in the direction of camera translation in space but does not supply the length of T. The actual translation vector T applied to the camera is a multiple of the vector $\vec{OF}$:

$$T = \lambda\vec{OF} = \lambda[x_f\,y_f\,f]^T, \qquad \lambda \in R. \quad (11)$$

In velocity-based models of 3-D motion, the FOE has frequently been interpreted as the *direction of instantaneous heading*, i.e., the direction of vehicle translation during an infinitely short period of time. When images are given as a sequence of snapshots taken at discrete instances of time with significant image motion in between, a discrete model seems more appropriate that treats the FOE as the direction of *accumulated* vehicle translation over a certain *period* of time.

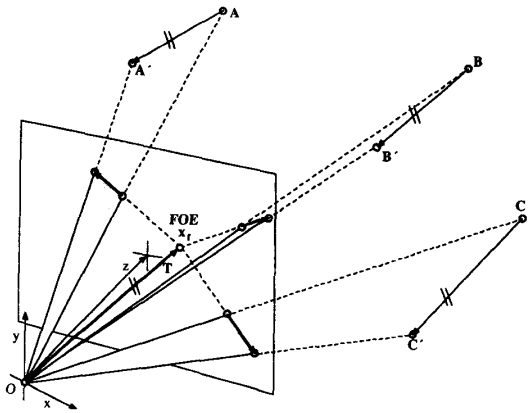

Fig. 3. Effects of camera translation in the forward direction. Under pure camera translation, points in the environment (A,B,C) move relative to the camera along parallel 3-D vectors in space. These parallels have a common vanishing point $x_f = (x_f\ y_f)$ (the focus of expansion, FOE), in the perspective image, from which all displacement vectors seem to expand.

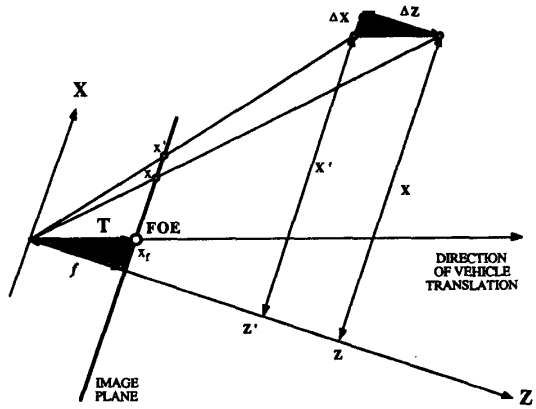

Fig. 4. Amount of expansion from the FOE for discrete time steps. The camera moves by a vector T in 3-D space, which passes through the lens center and the FOE in the image plane. The 3-D Z-axis is also the optical axis of the camera.

1) Measuring the Amount of Camera Translation: Fig. 4 shows the geometric relationships for measuring the amount of camera translation for the 2-D case. It can be considered as a top view of the camera, i.e., a projection onto the X/Z-plane of the camera-centered coordinate system. The cross section of the image plane is shown as a straight line. The camera is translating from left to right in the direction given by $\boldsymbol{T} = (x_f\ f)^T$. A stationary 3-D point is observed at two instances of time, which moves in space relative to the camera from $\boldsymbol{X}$ to $\boldsymbol{X}'$, resulting in two images x and x'.

$$\boldsymbol{X} = \begin{bmatrix} X \\ Z \end{bmatrix} \quad \text{and} \quad \boldsymbol{X}' = \begin{bmatrix} X' \\ Z' \end{bmatrix} = \begin{bmatrix} X - \Delta X \\ Z - \Delta Z \end{bmatrix}.$$

Using the inverse perspective transformation from (3) yields

$$Z = \frac{f}{x}X \quad \text{and} \quad Z' = Z - \Delta Z = \frac{f}{x'}X' = \frac{f}{x}(X - \Delta X).$$

From similar triangles (shaded in Fig. 4)

$$\frac{\Delta X}{x_f} = \frac{\Delta Z}{f},$$

and therefore

$$Z = \Delta Z \frac{x' - x_f}{x' - x} = \Delta Z\left(1 + \frac{x - x_f}{x' - x}\right). \tag{12}$$

Thus, the rate of expansion of image points from the FOE contains direct information about the distance of the corresponding 3-D points from the camera. Consequently, if the vehicle is moving along a straight line and the FOE has been located, the 3-D structure of the scene can be determined from the expansion pattern in the image. However, the distance Z of a 3-D point from the camera can only be obtained up to the scale factor ΔZ, which is the distance that the vehicle advanced along the Z-axis during the elapsed time.

When the velocity of the vehicle $(\Delta Z/t)$ in space is known, the absolute range of any stationary point can be computed. Alternatively, the linear velocity of the vehicle can be obtained if the actual range of a point in the scene is known (e.g., from laser range data). In practice, of course, any such technique requires that the FOE can be located in a small area, and that the observed image points exhibit significant expansion away from the FOE. As will be shown in the following section, image noise and camera distortion pose serious problems in the attempt to assure that both of the above criteria are met.

If a set of stationary 3-D points $\{(\boldsymbol{X}_i\boldsymbol{X}_i')\}$ is observed, then of course the translation in the Z-direction is the same for every point.

$$Z_i - Z_i' = Z_j - Z_j' = \Delta Z \quad \text{for all } i,j$$

Therefore, the range of every point is proportional to the observed amount of expansion of its image away from the FOE,

$$Z_i \propto \frac{x_i' - x_f}{x_i' - x_i},$$

which renders the relative 3-D structure of the set of points.

The effects of camera translation $\boldsymbol{T}$ can be formulated as a mapping $\boldsymbol{t}$ between two ordered sets of corresponding image points $\boldsymbol{I} = \{\boldsymbol{x}_i\}$ and $\boldsymbol{I}' = \{\boldsymbol{x}_i'\}$. Unlike in the case of pure camera rotation, this mapping not only depends upon the 3-D translation vector but also upon the position of each point in 3-D space. Therefore, the quantitative image effects of a particular translation $\boldsymbol{T}$ cannot be predicted without knowing the actual 3-D structure of the scene. However, there is an important *qualitative* property of the 2-D mapping $\boldsymbol{t}$, namely that each point maps onto a straight line passing through the original point and the FOE, i.e., if the vehicle undergoes pure forward translation, then there must exist one image location $\boldsymbol{x}_f$, such that $\boldsymbol{t}$ is a radial mapping between $\boldsymbol{I}$ and $\boldsymbol{I}'$ with respect to $\boldsymbol{x}_f$. In other words, the condition

$$\text{radial-mapping } (\boldsymbol{I},\boldsymbol{I}',\boldsymbol{x}_f):$$
$$\boldsymbol{t} = \{(\boldsymbol{x}_i,\boldsymbol{x}_i') \in \boldsymbol{I} \times \boldsymbol{I}' \mid \boldsymbol{x}_i' = \boldsymbol{x}_i + \mu_i(\boldsymbol{x}_i - \boldsymbol{x}_f),$$
$$\mu_i \in R, \mu_i \geq 0\}$$

must be satisfied. This observation is the key for decomposing the effects of arbitrary camera motion into its translational and rotational components.

III. Decomposition of Image Motion

The image effects of a composite 3-D camera motion $\boldsymbol{M}$: $\boldsymbol{X} \rightarrow \boldsymbol{X}' = \boldsymbol{R}_\phi\boldsymbol{R}_\theta(\boldsymbol{T} + \boldsymbol{X})$ can be summarized by a mapping $\boldsymbol{d} : \boldsymbol{I}_0 \rightarrow \boldsymbol{I}_1 = \boldsymbol{r}_\phi\boldsymbol{r}_\theta\boldsymbol{t}(\boldsymbol{I}_0)$, where $\boldsymbol{d}$ stands for *displacement* and

the mapping transforms the original image I_0 into the subsequent image I_1. For the purpose of clarity, we introduce the intermediate image $I^* = tI_0$ which is the result of the translation component of the vehicle's motion such that the condition *radial-mapping* (I_0, I^*, x_f) is satisfied for some x_f. Unlike the two images I_0 and I_1, this new image I^* is generally not observed, except when no camera rotation occurs. It serves as an intermediate result to be reached during the separation of the translational and rotational motion components.

Fig. 5 shows the top view of a vehicle traveling along a curved path to explain this decomposition. At the two instances in time t_0 and t_1, the position of the vehicle in space is specified by the location of a reference point (i.e., the lens center) P and the orientation Ω of the vehicle with respect to its environment. The original image I_0 is seen at time t_0. Following the adopted scheme of motion decomposition, the translation T is applied first, which takes the vehicle's reference point from position P_0 to position P_1 without changing its orientation Ω_0. This transforms image I_0 into image I^*. Notice that the FOE is found at the intersection of the 3-D translation vector T with the image plane I_0. In the second step, the vehicle is rotated by ω to its new orientation Ω_1, generating the final image I_1.

To allow a unique reconstruction of the actual motion parameters R_ϕ, R_θ, R_ψ, and T from their 2-D effects, a minimum number of environmental points must be included in the computation. Tsai and Huang [30] have shown that seven points in two perspective views suffice to obtain a unique interpretation in terms of rigid motion and structure, except for a few cases where points are arranged in some very special configuration in space. Ullman [31] reports computer experiments which indicate that six points are sufficient in many cases and seven or eight points yield unique interpretations in most practical cases. Of course, to reduce the effects of noise and tracking errors, in practice we will always try to include some redundancy by observing a much larger set of environmental points.

The fact that

$$t\,I_0 = I^* = r_\theta^{-1} r_\phi^{-1} I_1 \tag{13}$$

suggests two alternative strategies for separating the motion components.

1) FOE from Rotation: Successively apply combinations of inverse rotational mappings $\left(r_{\theta_1}^{-1} r_{\phi_1}^{-1}\right), \left(r_{\theta_2}^{-1} r_{\phi_2}^{-1}\right), \cdots, \left(r_{\theta_k}^{-1} r_{\phi_k}^{-1}\right)$ to the second image I_1, until the resulting image I' is a radial mapping with respect to the original image I_0. Then locate the FOE x_{f_k} in I_0.

2) Rotation from FOE: Successively select FOE-locations $x_{f_1}, x_{f_2}, \cdots, x_{f_n}$ (different directions of vehicle translation) in the original image I_0 and see if inverse rotational mappings $\left(\tilde{r}_{\theta_i}^{-1} \tilde{r}_{\phi_i}^{-1}\right)$ exist that yield a radial mapping with respect to the selected FOE x_{f_n} in the original image I_0.

Both alternatives were investigated under the assumption of realistic forward vehicle motion. Although originally the first approach had appeared more attractive, it turned out to be difficult to determine how close a given displacement field is to being radial when the location of the FOE is not given. In the presence of noise, this problem becomes even more difficult. The second approach was examined after it appeared that any method which extends the given set of displacement vectors *backwards* to find the FOE is inherently sensitive to image degradations [6].

In practice, the displacement vectors may not pass through a single pixel. Even for human observers it seems to be difficult to determine the exact direction of heading (i.e., the location of the FOE on the retina). Average deviation of human judgement from the real direction has been reported to be as large as 10° and up to 20° in the presence of large rotations [25]. It was, therefore, an important premise in this work that the final algorithm should determine an *area* of potential FOE-locations (called the *fuzzy FOE*) instead of a single (but probably incorrect) point. In the following, we briefly describe the FOE from rotation method and then present the details of our rotation from FOE technique.

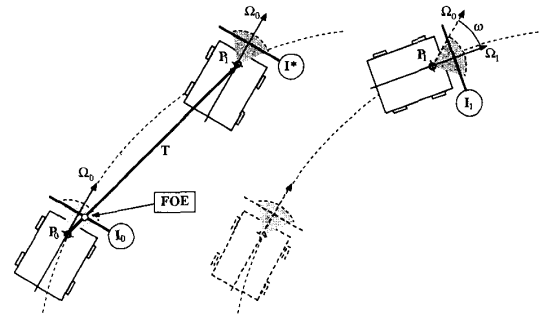

Fig. 5. Interpretation of the focus of expansion (FOE) for discrete time steps. Vehicle motion between its initial position (where image I_0 is observed) and its final position (image I_1) is modeled as two separate steps. (a) First the vehicle translates by a 3-D vector T from position P_0 to position P_1 without changing its orientation Ω_0. After this step, the intermediate image I^* would be seen. Subsequently (b), the vehicle rotates by changing its orientation from Ω_0 to Ω_1. Now image I_1 is observed. The FOE is found where the vector T intersects the image plane I_0 (and also I^*).

A. FOE from Rotation

In this method, the image motion is decomposed in two steps. First, the camera rotations are estimated and their inverse effects are applied to the second image I_1, thus producing a "derotated" image I'. If the rotation estimate was accurate, the resulting displacement field from I_0 to I' should be radial since only the effects of camera translation remain. The second step verifies that the displacement field is actually radial and determines the location of the FOE. In the FOE from rotation approach, two problems have to be solved:

1) How to estimate the rotational motion components without knowing the exact location of the FOE?
2) How to measure the "goodness of derotation" and locate the FOE?

There are several ways to get an initial estimate for the rotations. If the platform has sufficient inertia, the results from one pair of frames can be carried over to the following pair as an initial guess. Also, the displacement vectors of points at far distance from the camera (known from earlier observations) are not significantly affected by translation and can be used to obtain an immediate estimate for the rotations [e.g., using (10)]. Another approach is discussed by Bhanu and Burger [5], [6], where the range of possible rotations is successively constrained by geometric operations.

After applying the inverse rotations to the second image I_1, the question is how much the resulting displacement field between I_0 and the derotated image I' deviates from a radial mapping. Of course, due to finite image resolution, noise, and inaccuracies from point tracking, we can never expect any mapping to be perfectly radial. Prazdny [24] suggests to measure the disturbance of the displacement field by computing the variance of the intersections of one displacement vector with all other vectors. If those intersections lie close together, then the variance is small,

indicating that the displacement field is almost radial. Similarly, instead of selecting a particular displacement vector, one could compute the intersection with imaginary horizontal or vertical lines, as was pursued by Bhanu and Burger [5], [6].

Common to this class of techniques is that they all need to extend the given displacement vectors backwards along straight lines to find their intersections which, of course, multiplies the disturbances caused by any existing errors. This is particularly true for short displacement vectors. As a consequence, the error functions to measure the quantitative deviation from a radial displacement field are not well-behaved under noisy conditions. They usually exhibit local minima which prohibit efficient search for the optimal derotation (see Bhanu and Burger [5], [6] for details).

B. Rotation from FOE

While the above method iterates over rotation angles, the rotation from FOE technique, discussed in the following, successively evaluates potential FOE-locations. An initial *guess* for the FOE may be obtained from knowledge about the orientation of the camera relative to the vehicle. Subsequently, the solution from the previous frame pair can be used as a starting point. Once a particular FOE, $\boldsymbol{x}_f$, has been selected, the problem is to find the rotational mappings $\boldsymbol{r}_\theta^{-1}$ and $\boldsymbol{r}_\phi^{-1}$ which, when applied to the image $\boldsymbol{I}_1$ to produce $\boldsymbol{I}'$, will result in an optimal radial mapping between $\boldsymbol{I}_0$ and $\boldsymbol{I}'$ with respect to a given FOE, $\boldsymbol{x}_f$. To apply a local search strategy (e.g., the method of steepest descent), we need a suitable error function that is well behaved in a large region around the global minimum.

The error measure that was chosen for this purpose uses the deviation of the displacement vectors from straight rays originating from the selected FOE. Given a set of corresponding image points $\{(\boldsymbol{x}_i \boldsymbol{x}_i') \in \boldsymbol{I}_0 \times \boldsymbol{I}'\}$ and some FOE-location $\boldsymbol{x}_f$ the error measure $\boldsymbol{E}$ is defined as

$$\boldsymbol{E}(\boldsymbol{x}_f) = \sum_i \boldsymbol{E}_i = \sum_i d_i^2 = \sum_i \left[\frac{1}{|\boldsymbol{x}_i - \boldsymbol{x}_f|} (\boldsymbol{x}_i - \boldsymbol{x}_f) \times (\boldsymbol{x}_i' - \boldsymbol{x}_f) \right]^2. \quad (14)$$

Fig. 6 shows the interpretation of this error measure as the sum of the squared perpendicular distances of the displacement vectors' end points from the radial lines. Notice that this expression implicitly puts more weight upon the long (i.e., dominant) displacement vectors and less on short ones.

Since under perspective transformation, image points move along hyperbolic paths when the camera performs pure rotation, the resulting displacement is not uniform over the entire image plane (7). However, if the amount of rotation is small (less than 4°) or the focal length of the camera is large, we can replace the nonlinear mappings $\boldsymbol{r}_\theta^{-1}$ and $\boldsymbol{r}_\phi^{-1}$ by a linear shift vector $\boldsymbol{s}_{\phi\theta}$ which is independent of the image location, i.e.,

$$\boldsymbol{I}^* = \boldsymbol{r}_\theta^{-1} \boldsymbol{r}_\phi^{-1} \boldsymbol{I}_1 \approx \boldsymbol{s}_{\phi\theta} + \boldsymbol{I}_1 = \boldsymbol{I}^+. \quad (15)$$

Notice that this assumption does not imply that the algorithm is valid just for small rotations. It only serves as an approximation to estimate the optimal rotation angles more quickly. In most practical cases, this condition is satisfied, provided that the time interval between frames is sufficiently small. However, should the amount of vehicle rotation be very large for some reason, a coarse estimate of the actual rotation can be found and applied to the image before the FOE computation [6]. With $\boldsymbol{s}_{\phi\theta}$ as a free variable, the error measure (14) becomes

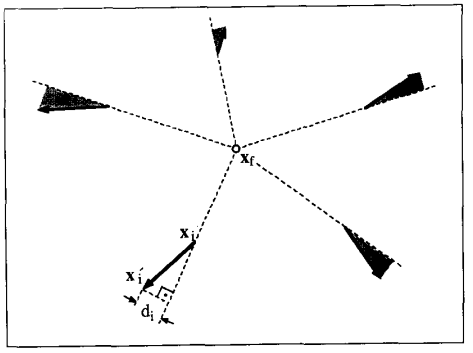

Fig. 6. Measuring the deviation from a radial displacement field. For the assumed FOE-location $\boldsymbol{x}_f$, d_i's are the perpendicular distances between the endpoints of the displacement vectors and radial lines emanating from $\boldsymbol{x}_f$ and passing through $\boldsymbol{x}_i$'s. The sum of the squared distances is used to quantify the deviation from a radial displacement field with respect to $\boldsymbol{x}_f$.

$$\boldsymbol{E}(\boldsymbol{x}_f, \boldsymbol{s}) \sum_i \left\{ \frac{1}{|\boldsymbol{x}_i - \boldsymbol{x}_f|^2} [(\boldsymbol{x}_i - \boldsymbol{x}_f) \times (\boldsymbol{x}_i' - \boldsymbol{x}_f + \boldsymbol{s}_{\phi\theta})]^2 \right\} \quad (16)$$

where $\boldsymbol{x}_i \in \boldsymbol{I}$ and $\boldsymbol{x}_i' \in \boldsymbol{I}'$.

This second-order error function can be minimized with standard numerical techniques to obtain an optimal value for $\boldsymbol{s}_{\phi\theta}$. To reduce this problem to a one-dimensional search, one point $\boldsymbol{x}_g$, called the *guiding point,* is selected in image $\boldsymbol{I}_0$ which is forced to maintain zero error for its displacement vector. Therefore, the corresponding point $\boldsymbol{x}_g^+$ must lie on a straight line passing through $\boldsymbol{x}_g^+$ and $\boldsymbol{x}_f$. Any shift $\boldsymbol{s}$ (see Fig. 7) applied to the set of image points $\boldsymbol{x}' \in \boldsymbol{I}'$ must keep $\boldsymbol{x}_g^+$ on this straight line, i.e., $\boldsymbol{x}_g^+ = \boldsymbol{x}_g' + \boldsymbol{s} = \boldsymbol{x}_f + \lambda(\boldsymbol{x}_g - \boldsymbol{x}_f)$ for all $\boldsymbol{s}$, and thus,

$$\boldsymbol{s} = \boldsymbol{x}_f - \boldsymbol{x}_g' + \lambda(\boldsymbol{x}_g - \boldsymbol{x}_f), \qquad \lambda \in R. \quad (17)$$

For example, for $\lambda = 1$, $\boldsymbol{s} = \boldsymbol{x}_g - \boldsymbol{x}_g'$ which is the negative of the displacement vector from $\boldsymbol{x}_g$ to $\boldsymbol{x}_g'$. In this case, $\boldsymbol{x}_g^+$ and $\boldsymbol{x}_g$ would overlap.

This leaves λ as the only free variable and the error function (16) becomes

$$\boldsymbol{E}(\lambda) = \sum_i [\lambda A_i - B_i + C_i]^2 \quad (18)$$

with

$$A_i = \frac{1}{r_i}[(x_i - x_f)(y_g - y_f) - (y_i - y_f)(x_g - x_f)]$$

$$B_i = \frac{1}{r_i}(y_i - y_f)(x_i' - x_g')$$

$$C_i = \frac{1}{r_i}(x_i - x_f)(y_i' - y_g')$$

$$r_i = \sqrt{(x_i - x_f)^2 + (y_i - y_f)^2}.$$

Differentiating (18) with respect to λ and setting the result equal to zero yields the parameter λ_{opt} for the optimal shift $\boldsymbol{s}_{\text{opt}}$ as

$$\lambda_{\text{opt}} = \frac{\Sigma A_i B_i - \Sigma A_i C_i}{\Sigma A_i^2}.$$

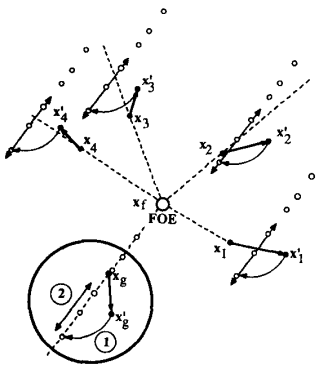

Fig. 7. Use of a guiding point. The problem is to register the two images $I = \{x_i\}$ and $I' = \{x'_i\}$ onto a radial pattern out of the FOE x_f (dashed lines) by shifting the second image I' uniformly by some (unknown) 2-D vector s. We simplify this to a 1-D search by selecting a displacement vector $x_g \rightarrow x'_g$ whose deviation from the radial line must be zero under any shift vector s. Intuitively, this means that the entire image I' is first shifted such that x'_g coincides with a radial line through x_g (1) and then I' is translated parallel to this line (2) until a minimum error is reached.

The optimal shift vector s_{opt} is obtained using (17). However, to evaluate the selected FOE-location x_f, we are primarily interested in the error value $E(\lambda_{\text{opt}})$ that would result from applying the approximate derotation s_{opt}. $E(\lambda_{\text{opt}})$ is obtained by inserting λ_{opt} into (18), giving

$$E(\lambda_{\text{opt}}) = \lambda_{\text{opt}}^2 \sum A_i^2 + 2\lambda_{\text{opt}}\left(\sum A_i C_i - \sum A_i B_i\right) - 2\sum B_i C_i + \sum B_i^2 + \sum C_i^2, \quad (19)$$

or

$$E(\lambda_{\text{opt}}) = -\frac{(\Sigma A_i B_i - \Sigma A_i C_i)^2}{(\Sigma A_i^2)} - 2\sum B_i C_i + \sum B_i^2 + \sum C_i^2. \quad (20)$$

The normalized error E_N shown in the results that follow (Figs. 9–15) is defined as

$$E_N = \sqrt{\frac{1}{N} E(\lambda_{\text{opt}})}, \quad (21)$$

where N is the number of displacement vectors used for computing the FOE.

Since in a displacement field caused by pure camera translation all vectors must point away from the FOE, this restriction must hold for any candidate FOE-location (Fig. 8). If after applying $s_{\text{opt}}(x_f)$ to the second image I', the resulting displacement field contains vectors pointing *toward* the hypothesized x_f, then this FOE-location is *prohibited* and can be discarded from further consideration. Fig. 8 shows a field of five displacement vectors. The optimal shift s_{opt} for the given x_f is shown as a vector in the lower right-hand corner. When s_{opt} is applied to point x'_1, the resulting displacement vector (shown with a heavy line) does not point away from the FOE. Since its projection onto the line $\overline{x_f x_1}$ points *toward* the FOE, it is certainly not consistent with a radial expansion pattern.

The following function *Evaluate_Single_FOE* examines one hypothetical FOE-location x_f in a given pair of images I_0 and I_1. It uses the functions *Optimal_Shift* for computing the optimal shift vector s_{opt} and *Equivalent_Rotation* to obtain the rotation

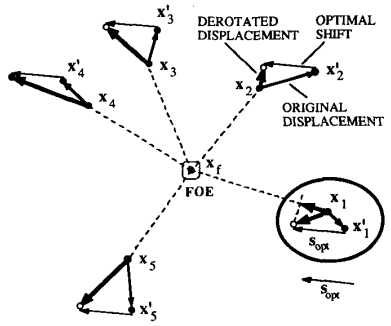

Fig. 8. FOE-locations are *prohibited* if the displacement field resulting from the application of the optimal shift s_{opt} contains vectors pointing towards the FOE. This is the case at point x_1.

angles equivalent to s_{opt} [by (10)]. The parameters $\phi_{\max}$ and $\theta_{\max}$ are the maximum angles of rotation for which the approximation in (15) is acceptable. If the estimated rotation angles exceed that limit, the intermediate image I^+ is derotated *exactly* by procedure *Derotate_Image* [using (6)] before another estimate is computed.

Evaluate_Single_FOE$(x_f, I_0, I_1, \phi_{\max}, \theta_{\max})$:
  $I^+ \leftarrow I_1$;
  $(\phi, \theta) \leftarrow (0, 0)$;
  repeat /* usually only one iteration required */
    $(s_{\text{opt}}, error) \leftarrow$ *Optimal_Shift*(I_0, I^+, x_f);
    $(\phi^+, \theta^+) \leftarrow$ *Equivalent_Rotation* (s_{opt});
    $(\phi, \theta) \leftarrow (\phi, \theta) + (\phi^+, \theta^+)$;
    $I^+ \leftarrow$ *Derotate_Image* (I_1, ϕ, θ);
  until $(\phi^+ \le \phi_{\max}$ & $\theta^+ \le \theta_{\max})$
  return $(I^+, \phi, \theta, error)$.

The error function E is computed in time proportional to the number of displacement vectors N. The final size of the FOE-area depends on the local shape of the error function and can be constrained not to exceed a certain maximum M. Therefore, the time complexity is $O(MN)$.

The first set of experiments was conducted on synthetic imagery to investigate the behavior of the error measure under various conditions, namely

- the average length of the displacement vectors (longer displacement vectors lead to a more accurate estimate of the FOE),
- the amount of residual rotation components in the image, and
- the amount of noise applied to the location of image points.

Fig. 9 shows the distribution of the normalized error E_N for a sparse and relatively short displacement field (*length factor* = 2 = 8 pixels) containing 7 vectors. Residual rotation components of $\pm 2°$ in horizontal and vertical direction are present in Fig. 9(b)–(d) to visualize their effects upon the image. The displacement vector through the *guiding point* is marked with a heavy line. The choice of this point is not critical, but it should be located at a considerable distance from the FOE to reduce the effects of noise upon the direction of the vector $\vec{x_f x_g}$. In Fig. 9, the error function is sampled in a grid with a width of 10 pixels over an area of 200 by 200 pixels around the actual FOE, which is marked by a small square. The size of the circle at each location indicates the amount of error, i.e., the deviation from the radial displacement field that would result if that location were picked as the FOE. Heavy circles

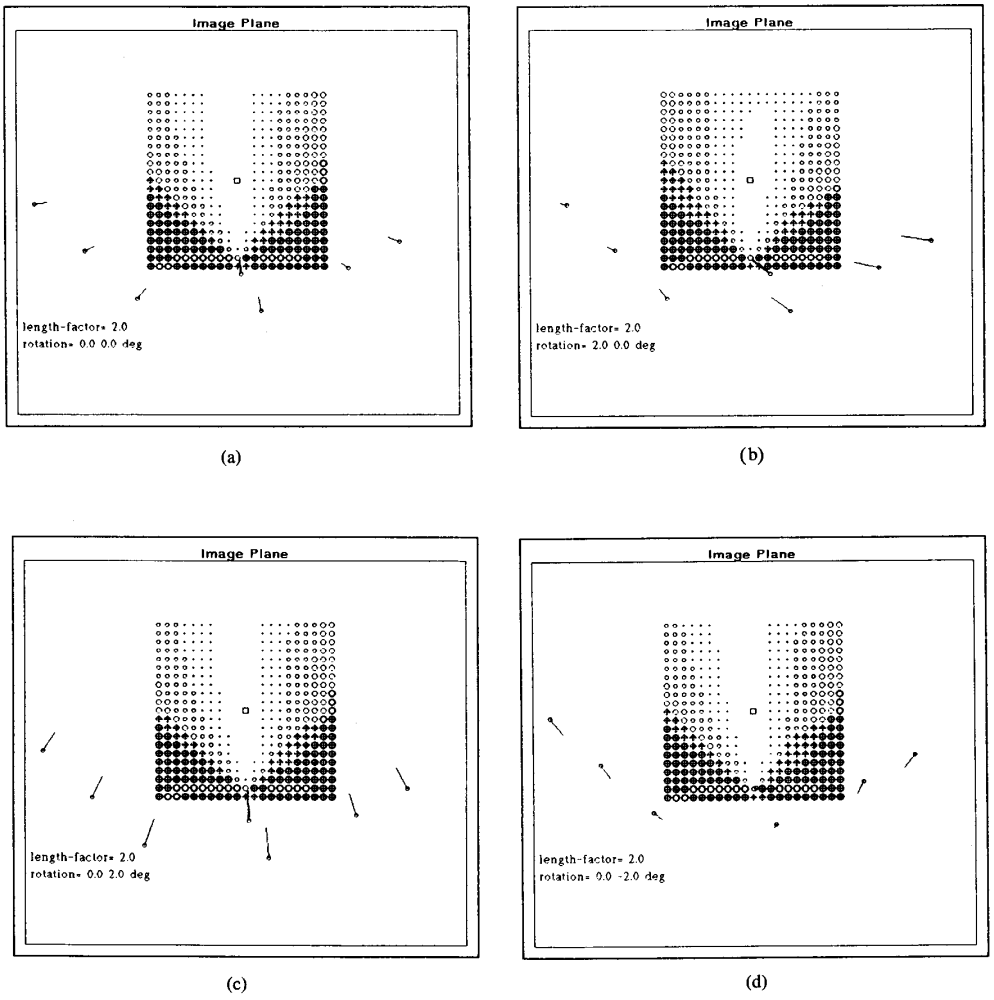

Fig. 9. Error function for a synthetic displacement field. Displacement field and minimum error at selected FOE-locations. The shape of the error function is plotted over an area of ±100 pixels around the actual FOE (marked with a small square). The diameter of each circle reflects the amount of normalized error (21) for that particular FOE-location. Heavy circles indicate error values above a certain threshold (4.0), *prohibited* locations (as defined earlier) are marked "+". (a) No residual rotation. (b) 2.0° of horizontal camera rotation (camera rotated to the left). (c) 2.0° vertical rotation (camera rotated upwards). (d) −2.0° vertical rotation (camera rotated downwards).

indicate error values which are above a certain threshold (4.0). Those FOE-locations that would result in displacement vectors which point *toward* the FOE (as described earlier) are marked as prohibited (+). It can be seen that this two-dimensional error function is smooth and monotonic within a large area around the actual FOE (marked by a small square). The shape of this error function makes it possible that, even with a poor initial guess, the global optimum can be found by local search methods.

Figs. 10 to 15 show the effects of various conditions upon the behavior of this error function in the same 200 × 200 pixel square around the actual FOE as in Fig. 9.

An important criterion is the function's behavior when the amount of camera translation is small or the displacement vectors are noisy. Fig. 10 shows the effects of varying the average length of the displacement vectors in the range of 4–60 pixels in the absence of any residual rotation or noise (except digitization noise). Note that longer displacement vectors result in a sharper minimum around the actual FOE.

Fig. 11 shows the effect of increasing residual rotation in horizontal direction upon the shape of the error function: Fig. 12 shows the effect of residual rotation in vertical direction. Here, it is important to notice that the displacement field used is extremely nonsymmetric along the Y-axis of the image plane. This is motivated by the fact that in real ALV images, long displacement vectors are most likely to be found from points on the ground, which are located in the lower portion of the image. Therefore, positive and negative vertical rotations have been applied in Fig. 12.

In Fig. 13, residual rotations in both horizontal and vertical direction are present. It can be seen [Fig. 13(a)–(e)] that the error function is quite robust against rotational components in the image. The result in Fig. 13(e) shows the effect of large combined rotation of 4.0°/4.0° in both directions. Here, the minimum of

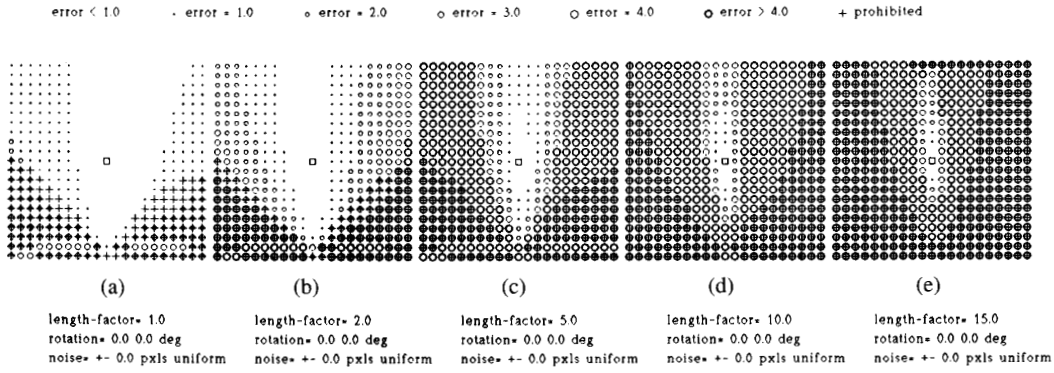

Fig. 10. Effects of increasing the average length of displacement vectors upon the shape of the error function. The same displacement field as in Fig. 9 was used. (a) Average length is 4 pixels, (b) 8 pixels, (c) 20 pixels, (d) 40 pixels, (e) 60 pixels. (Length factor varies from 1 to 15.) Note that FOE is better defined by longer displacement vectors.

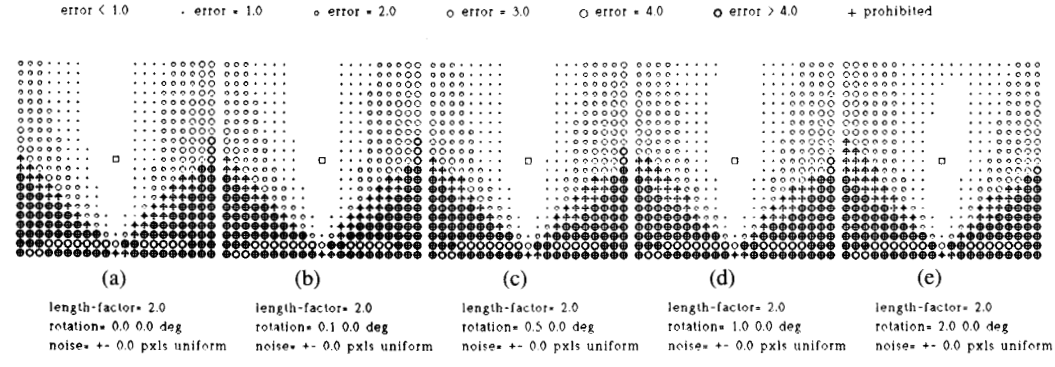

Fig. 11. Effects of increasing residual rotation in horizontal direction upon the shape of the error function for relatively short vectors (length factor 2.0). No noise was applied.

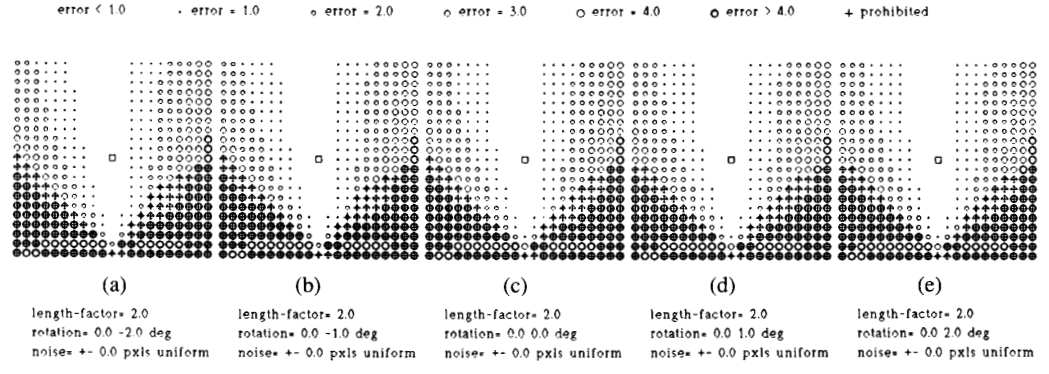

Fig. 12. Effects of increasing residual rotation in vertical direction upon the shape of the error function for relatively short vectors (length factor 2.0). No noise was applied.

the error function is considerably off the actual location of the FOE because of the error induced by using a linear shift to approximate the non-linear derotation mapping. In such a case, it would be necessary to actually *derotate* the displacement field by the amount of rotation equivalent to s_{opt} found at the minimum of this error function and repeat the process with the derotated displacement.

The effects of various amount of noise are shown in Fig. 14. For this purpose, a random amount (with uniform distribution) of displacement was added to the original (continuous) image location and then rounded to integer pixel coordinates. Random displacement was applied in ranges from ±0.5 to ±4.0 pixels in both horizontal and vertical direction. Since the displacement field contains only seven vectors, the results do not provide

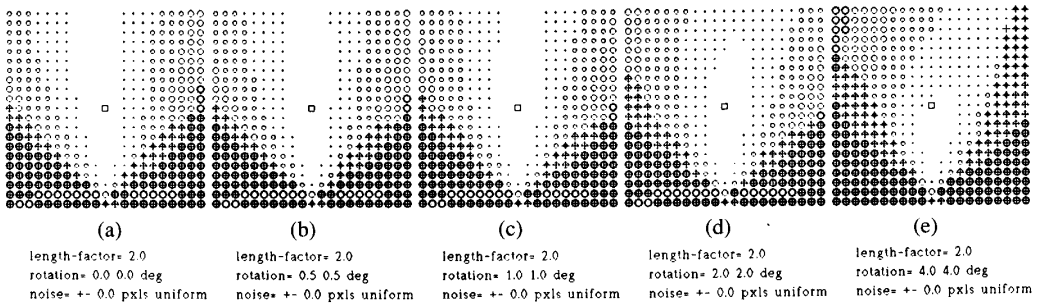

Fig. 13. Effects of increasing residual rotation in horizontal and vertical direction upon the shape of the error function for relatively short vectors (length factor 2.0). No noise was applied.

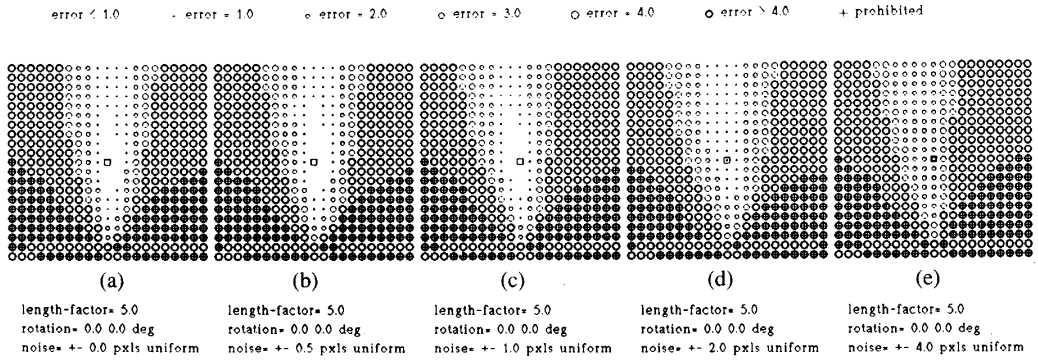

Fig. 14. Effects of varying the amount of noise. Uniform noise was applied to the same displacement field as in Fig. 9. (a) Zero noise, (b) ±0.5 pixels, (c) ±1.0 pixels, (d) ±2.0 pixels, (e) ±4.0 pixels. The error function becomes flat with increasing noise levels.

information about the statistical effects of image noise. This would require more extensive modeling and simulation. However, this figure serves as an indicator for the amount of noise present in the image and the reliability of the final result. It can be observed that the error function flattens out with increasing levels of noise, although the generic shape of the function does not change. Further, the location of the global minimum error may be located considerably off the actual FOE which makes it difficult to locate the FOE precisely under noisy conditions.

Again, it should be noted that the length of the displacement vectors is an important factor. The shorter the displacement vectors are, the more difficult it is to locate the FOE correctly in the presence of noise. Fig. 15 shows the error functions for two displacement fields with different average vector lengths. For the shorter displacement field (*length-factor* 2.0) in Fig. 15(a), the shape of the error function changes dramatically [compare Fig. 13(a)]. A search for the minimum error would inevitably converge toward a point indicated by the small arrow, far off the actual FOE. For the image with *length-factor* 5.0 [Fig. 15(b)], the minimum of the error function coincides with the actual location of the FOE (a). The different result for the same constellation of points in Fig. 14(d) is caused by the different random numbers (noise) obtained in each experiment. This experiment shows that a sufficient amount of displacement between consecutive frames is essential for reliably determining the FOE and thus, the direction of vehicle translation.

These experiments demonstrate the need to somehow quantify the reliability of the resulting FOE by analyzing the local shape of the error function. Our goal is therefore not to compute a single FOE, but rather a 2-D distribution of possible FOE-locations. The final result is a connected image region, which we call a "Fuzzy FOE," that can be assumed to contain the actual FOE with high certainty. The following algorithm describes the basic steps in computing the Fuzzy FOE for a given pair of images $\boldsymbol{I}_0$ and $\boldsymbol{I}_1$:

Fuzzy_FOE($\boldsymbol{I}_0, \boldsymbol{I}_1$):

1) Guess an initial FOE, $\boldsymbol{x}_0$.
2) Starting from $\boldsymbol{x}_0$, search for an FOE, $\boldsymbol{x}_{\min}$, with minimum error (following the steepest descent).
3) Around $\boldsymbol{x}_{\min}$, grow a connected region such that the error for each member FOE location is below a given limit.

In implementing this algorithm, several measures can be taken to improve its efficiency. First, the required search in steps 2) and 3) can be done over a grid of varying resolution from coarse to fine. To eliminate multiple evaluations of the same FOE-location (by function *Evaluate_Single_FOE*), results could be kept for later reference, e.g., in a hash table. Finally, FOE-locations can be evaluated in parallel if suitable hardware is available. Results from computing the Fuzzy FOE on real data taken from a moving ALV are shown in the following section.

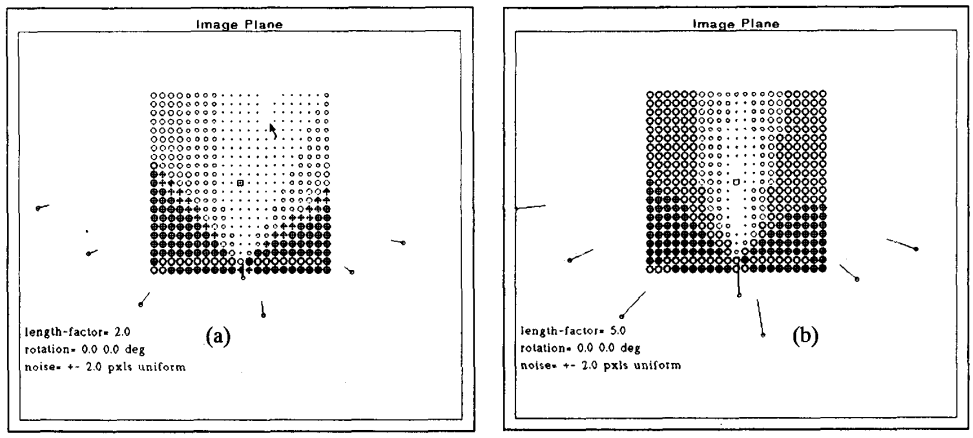

Fig. 15. The effects of uniform noise applied to image point coordinates for different average vector lengths (length factors 2.0 and 5.0). For the short displacement field (a) the disturbance moves the local minimum (arrow) far off the actual FOE. The same amount of noise applied to the longer displacement field has much less dramatic effects.

IV. Experimental Results

In the following, the results of the FOE-algorithm and computation of the vehicle's velocity over ground (see Appendix) are shown on a real image sequence, illustrated in Fig. 16, taken from the moving ALV. The spatial resolution in these images is 512 × 512 pixels. The vehicle traveled at approximately 14 kilometers per hour and the elapsed time between each pair of frames is 0.5 seconds, such that the translation vector has a length of about 1.9 meters.

For the experiment described in the following, points were tracked manually after computing binary edge images from the original sequence. Since some information (e.g., color cues) that is useful for a human observer is lost in the process of edge detection, the potential performance of an automatic tracking procedure should not be too far from these results. The assumption was that points that are significant in an edge image should also be relatively easy to track by a program, as compared to points which are subjectively selected in a high-quality color image. Recent experiments [7], [18], in fact, indicate that adequate results can be obtained by selecting and tracking point features in a fully automatic mode.

Approximately 25 points were selected in each image and the period of observation for these points was in the range of 2–16 frames. The particular types of local image features considered were endpoints and corners of lines as well as centers of small closed regions. Features on the road surface, which are important for estimating camera motion, turned out to be difficult to follow from frame to frame when they are approached by the vehicle. Since the edge operator uses the same mask regardless of the 3-D distance of the feature, the resulting edge image may change dramatically when features get closer and change their scale. A successful implementation for feature extraction and tracking will need to take this into account and employ some form of range-dependent image operation [7], [18]. Fig. 17 shows 16 edge images of the original ALV sequence (Fig. 16), where the selected feature points are labeled with numbers.

Fig. 18 shows the final results of computing the vehicle's motion for the sequence in Fig. 16. Each frame n at time t displays the motion estimates for the period between t and the previous frame $n-1$ at time $t-1$. Therefore, the first motion estimate is available after the second frame ($n = 183$). Throughout, in search for the FOE, the optimal FOE-location (i.e., the one with minimum error) from the previous frame pair was taken as the initial guess. For the very first pair of frames, the FOE was guessed from the known camera orientation relative to the vehicle.

Each motion estimate consists of three components: a) the fuzzy FOE, b) the angles of horizontal and vertical rotation, and c) the approximate distance traveled over ground. The fuzzy FOE is marked by a shaded area of varying shape and size. The jagged outline of the fuzzy FOE is caused by the relatively coarse grid (10 pixels wide) used for growing the region. The small circle inside this region marks the location of the optimal FOE. The shape of the FOE-region depends strongly upon the most dominant (longest) displacement vectors in the scene. Since these vectors are usually found in the lower central parts of the image, the FOE tends to be elongated along the vertical axis, i.e., the horizontal position of the FOE is usually better defined than its vertical position. This may be different, of course, for other types of scenes.

The estimates for the rotations in horizontal and vertical direction are shown in a coordinate frame ±1° in the lower left-hand corner of each image. Since the amount of rotation is relatively small, it was never necessary to apply intermediate derotation during the FOE search. Along with the original displacement vectors (solid lines), the vectors obtained after derotation (dotted lines) are also shown.

The absolute velocity of the vehicle is estimated after computing the FOE (see Appendix). The essential measure used for this calculation is the height of the camera above the ground, which is constant and known (3.3 meters). If the road is assumed to be flat and parallel to the direction of vehicle translation, the 3-D distances of points on the road surface and thus the amount of vehicle translation can be found in absolute terms. The estimated advancement (in meters) between each frame pair is also given in Fig. 18. Since points on the road, particularly those close to the camera, are difficult to track, these estimates are approximate.

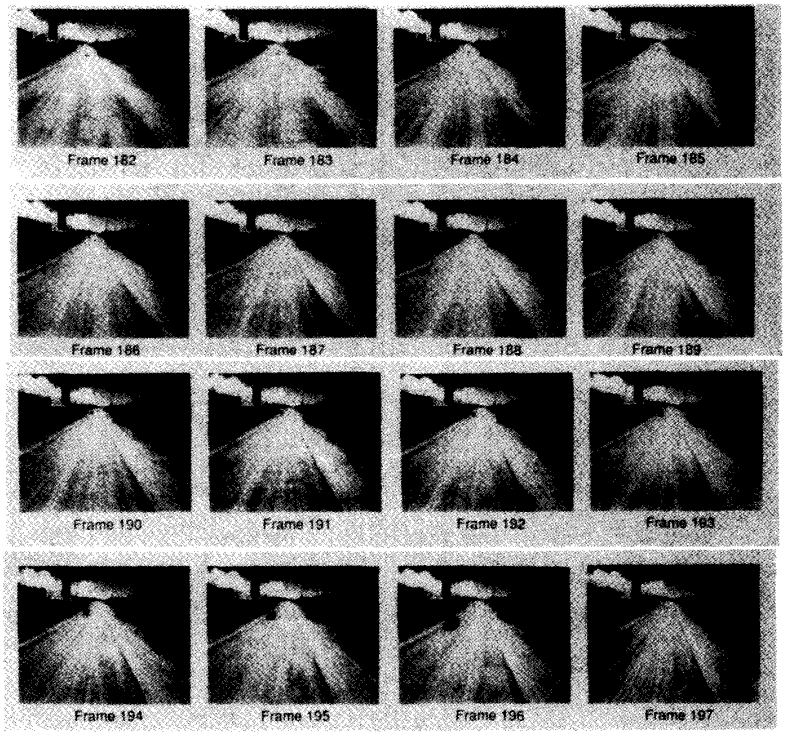

Fig. 16. A sequence (frames 182–197) taken from a moving autonomous land vehicle (ALV). The scene contains two moving objects, one car moving away from the ALV and another car approaching the ALV.

V. Conclusions

The goal of this work was to develop a robust technique for computing the parameters of self-motion of a land vehicle from visual information. This information is useful for a variety of tasks, such as vehicle control, navigation, obstacle avoidance, etc. It turned out that particularly the direction of heading (i.e., the focus of expansion—FOE) is difficult to compute under practical conditions, i.e., finite image resolution, noise, and feature tracking errors.

Our solution to this problem is not to search for a single-point FOE, but rather to determine a 2-D region of possible FOE-locations (termed the fuzzy FOE), whose shape is an explicit indicator for the reliability of the result. It is particularly suited for motion sequences that exhibit a significant translation component, i.e., for most vehicular applications. In contrast to other methods, however, the fuzzy FOE performs well even under the conditions of small rotation, image noise and also provides a measure for the quality of the result. Experiments show that even erroneous point matches in the given image displacement vector field can be tolerated, which is absolutely necessary in a fully automated process.

Although land-based vehicles and autonomous robots have been considered as the main application, the approach appears to be useful in other environments as well. For example, the described algorithm has been successfully applied to images taken from a helicopter flying at low altitude [7], where the rotations are much more significant than in the examples shown here.

Appendix
Computing Velocity Over Ground

In the following, it is shown how the absolute velocity of the vehicle can be estimated after the location of the FOE has been determined. The essential measure used for this calculation is the absolute height of the camera above the ground which is constant and known. As discussed in Section II, from the derotated displacement field and the location of the FOE, the 3-D layout of the scene can be obtained up to a common scale factor (12). This scale factor and, consequently, the velocity of the vehicle can be determined if the 3-D position of one point in space is known. Furthermore, it is easy to show [16] that it is sufficient to know only one coordinate value of a point in space to reconstruct its position in space from its location in the image.

As the ALV travels on a fairly flat surface, the road can be approximated as a plane which lies parallel to the vehicle's direction of translation (see Fig. 19). This approximation holds at least for a good part of the road in the field of view of the camera. Since the absolute height of the camera above the ground is constant and known, it is possible to estimate the positions of points on the road surface with respect to the vehicle in *absolute* terms. From the changing distances between these points and the camera, the actual advancement and speed can be determined.

First, a new coordinate system is introduced which has its origin in the lens center of the camera. The Z-axis of the new system passes through the FOE in the image plane and points, therefore, in the direction of translation. The original camera-centered coordinate system (XYZ) is

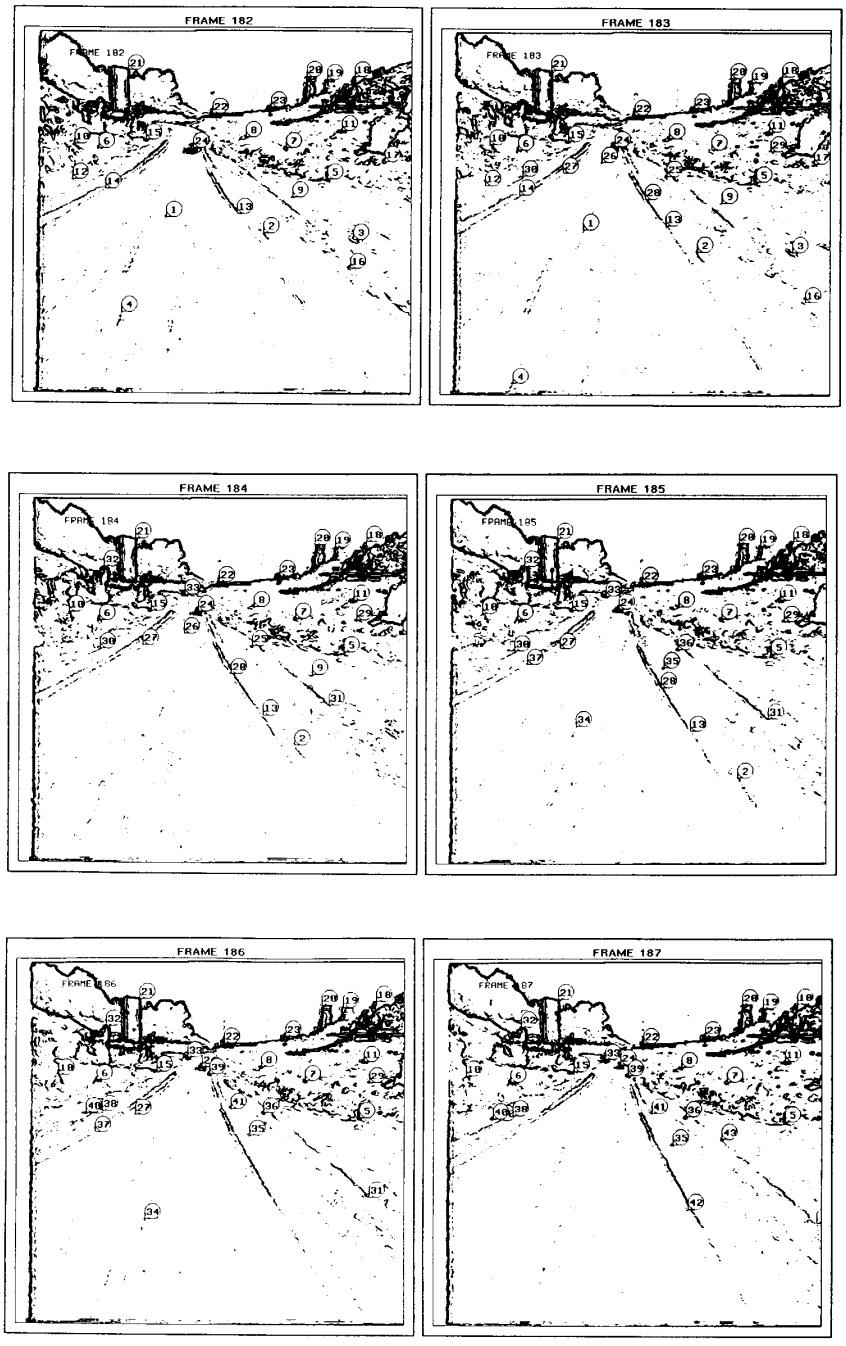

Fig. 17. ALV image sequence (frames 182–197) after edge detection and point selection. Points were tracked manually on these images from one frame to the next and labeled with numbers. The actual image location of each point lies at the lower left-hand corner of the corresponding label.

transformed into the new frame $(X'\,Y'\,Z')$ merely by applying horizontal and vertical rotation until the Z-axis lines up with the FOE.

The horizontal and vertical orientation in terms of *pan* and *tilt* are obtained by "rotating" the FOE $(x_f\,y_f)$ into the center of the image (0 0) using (10):

$$\theta_f = -\tan^{-1}\frac{x_f}{f}, \quad \phi_f = -\tan^{-1}\left(y_f \frac{f^2}{(f^2+x_f^2)f^2 - x_f^2 y_f^2}\right).$$

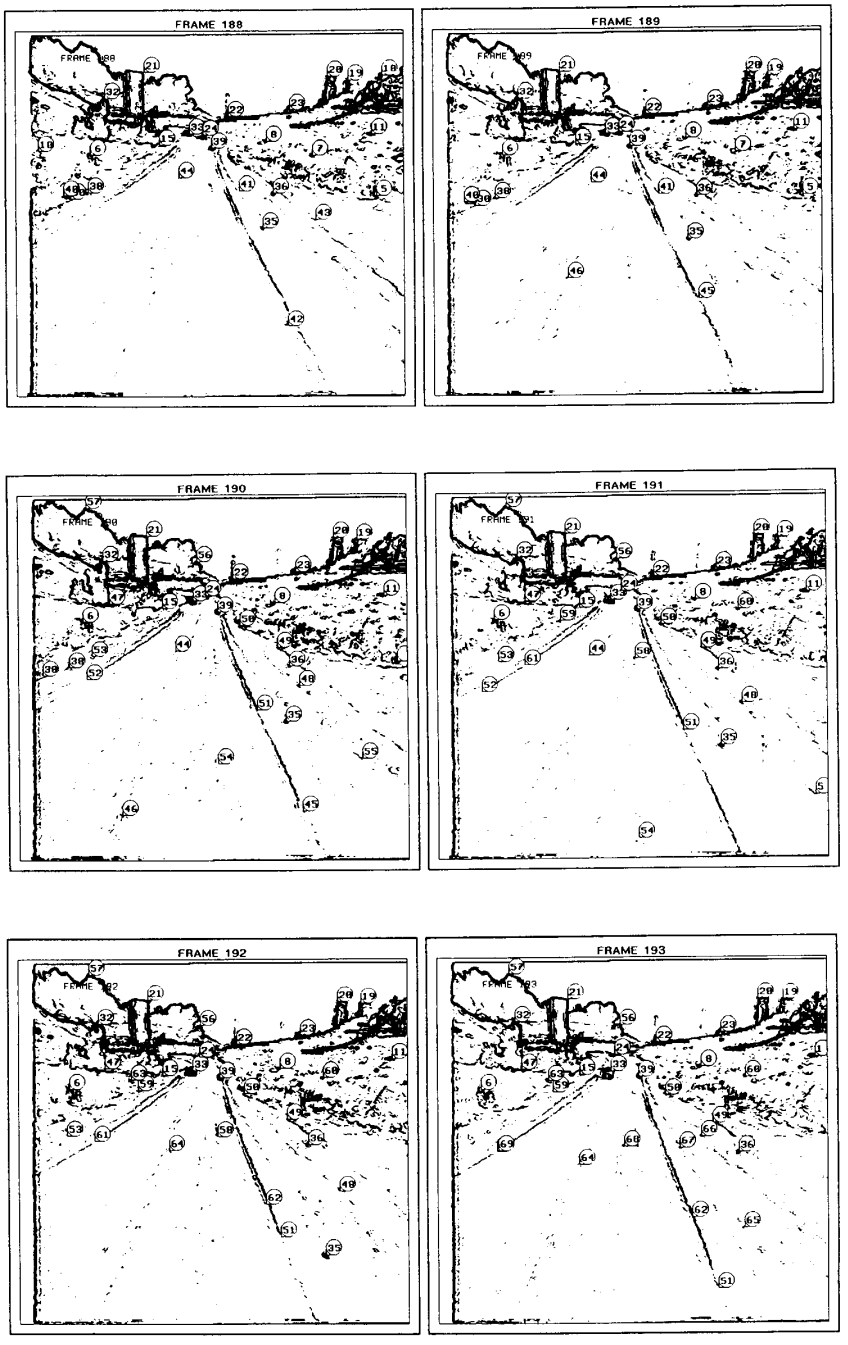

Fig. 17. (*Continued.*)

The two angles θ_f and ϕ_f represent the orientation of the camera in 3-D with respect to the new coordinate system. This allows us to determine the 3-D orientation of the projecting rays passing through image points by use of the inverse perspective transformation. A 3-D point $\boldsymbol{X}$ in the environment whose image $\boldsymbol{x} = (xy)$ is given, lies on a straight line in space defined by

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \kappa \begin{bmatrix} \cos\theta_f & \sin\theta_f \sin\phi_f & -\sin\theta_f \cos\phi_f \\ 0 & \cos\phi_f & \sin\phi_f \\ \sin\theta_f & -\cos\theta_f \sin\phi_f & \cos\theta_f \cos\phi_f \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix}.$$

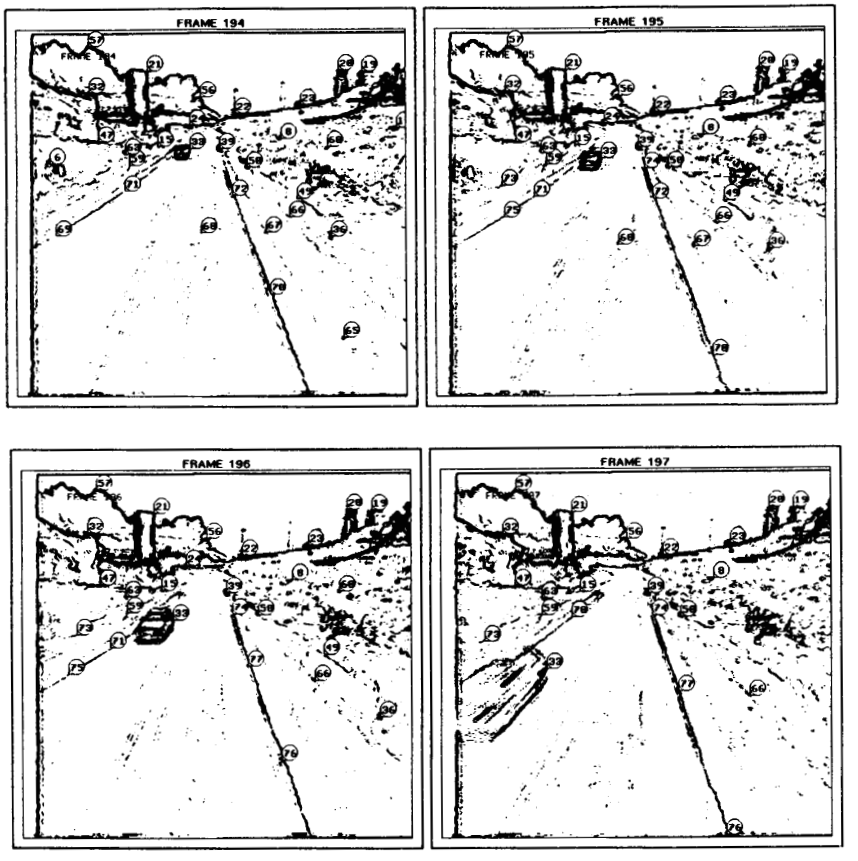

Fig. 17. (*Continued.*)

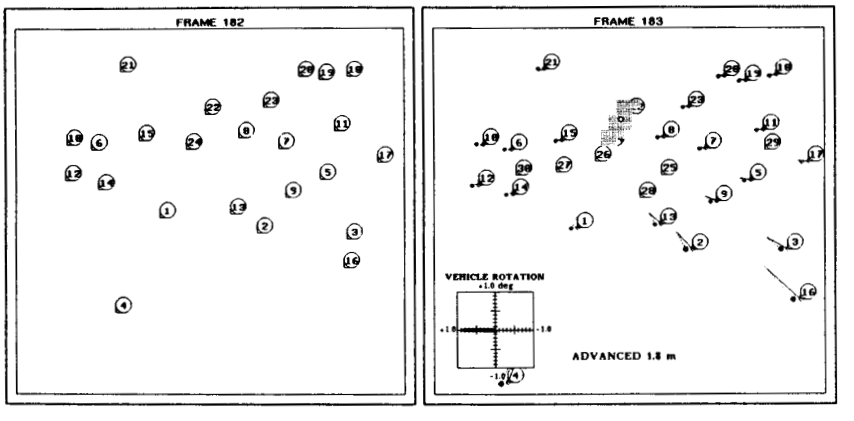

Fig. 18. Displacement vectors and estimates of vehicle motion for the ALV image sequence shown in Fig. 16. The shaded area in each image marks the computed fuzzy FOE, whose shape depends upon the most dominant (i.e., longest) displacement vectors. The small circle inside the shaded area is the FOE-location with minimum error. Estimated vehicle rotation is plotted in a coordinate frame in the range over ±1.0°. The estimated absolute advancement for each frame pair is given in meters. Original and derotated displacement vectors are drawn with solid and dotted lines, respectively.

For points on the road surface, the Y-coordinate is $-h$ which is the height of the camera above ground. Therefore, the value of κ_s for a point on the road surface (x_s, y_s) can be estimated as

$$\kappa_s = \frac{-h}{y_s \cos\theta_f + f \sin\theta_f}$$

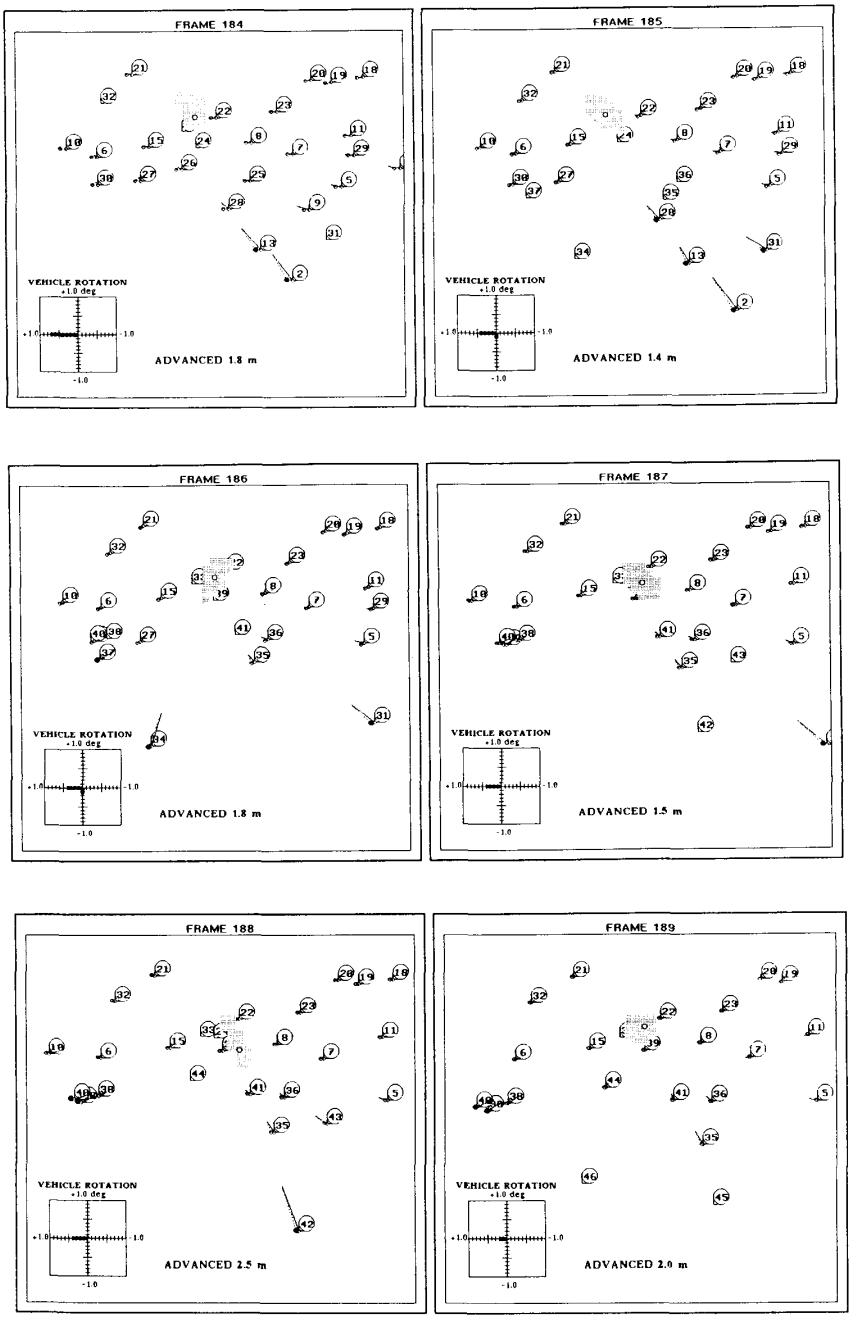

Fig. 18. (*Continued.*)

and its 3-D distance is found by inserting v_s into the above equation as

$$Z_s = -h\frac{x_s \sin\theta_f - y_s \cos\theta_f \sin\phi_f - f\cos\theta_f \cos\phi_f}{y_s \cos\phi_s + f\sin\phi_s}.$$

If a point on the ground is observed at two instances of time, x_s at time t and x'_s at t', the resulting distances from the vehicle Z_s at t and Z'_s at t' yield the amount of advancement $\Delta Z_s(t,t')$ and estimated velocity $V_s(t,t')$ in this period as

$$\Delta Z_s(t,t') = Z_s - Z'_s, \qquad V_s(t,t') = \frac{Z_s - Z'_s}{t' - t}.$$

Of course, image noise and tracking errors have a large impact upon the quality of the final velocity estimate. Therefore, the

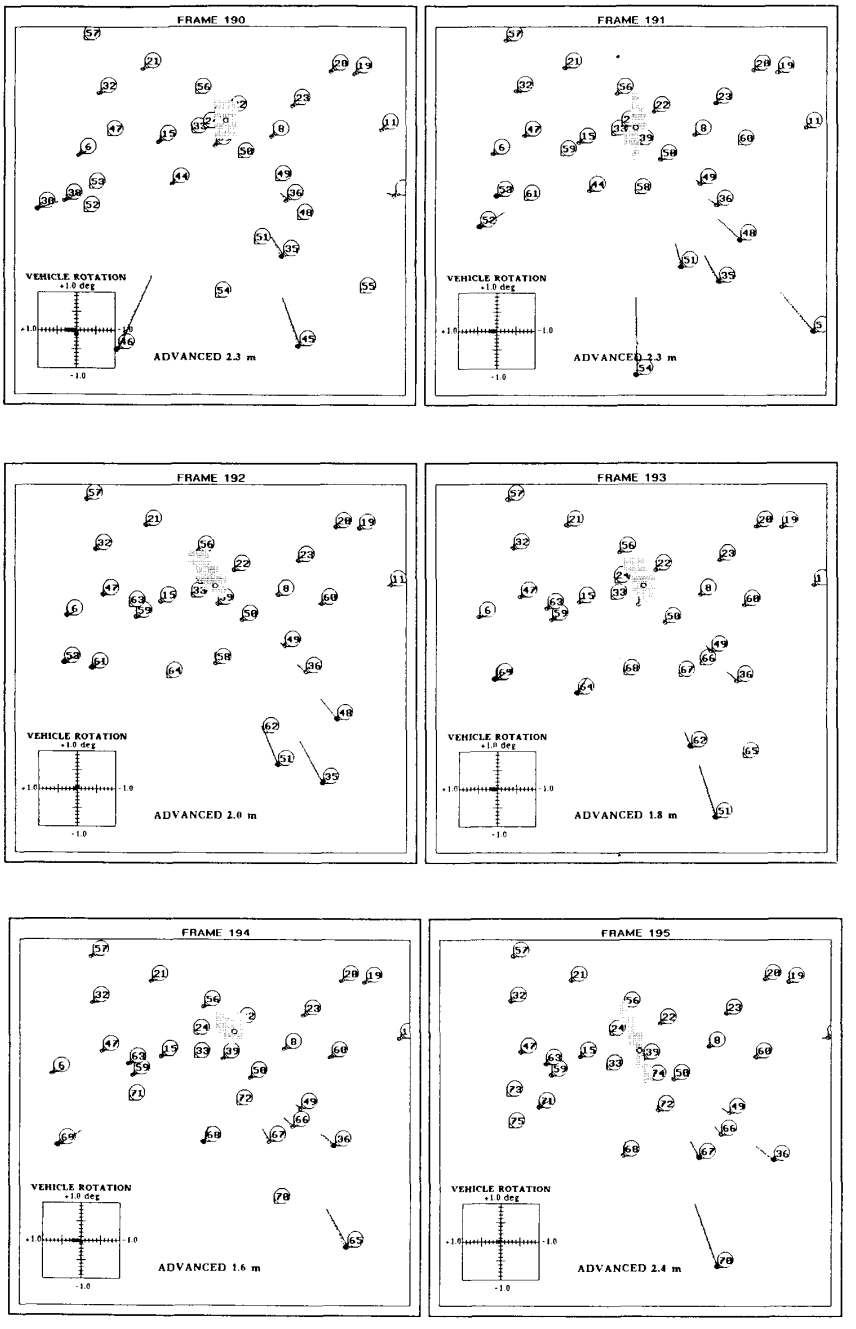

Fig. 18. (*Continued.*)

longest available displacement vectors are generally selected for this measurement, i.e., those vectors which are relatively close to the vehicle. Also, in violation of the initial assumption, the ground surface is never perfectly flat. In order to partially compensate these errors and to make the velocity estimate more reliable, the results of the measurements on individual vectors are combined. The length of each displacement vector $|\boldsymbol{x}_i - \boldsymbol{x}_i'|$ in the image is used as the weight for its contribution to the final result. Given a set of suitable displacement vectors $S = \{\boldsymbol{x}_i - \boldsymbol{x}_i'\}$, the estimate of the distance traveled by the vehicle is taken as the weighed average of the measurements ΔZ_i on individual vectors

$$\tilde{\Delta} Z(t, t') = \frac{\Sigma\left(|\boldsymbol{x}_i - \boldsymbol{x}_i'| \Delta Z_i\right)}{\Sigma |\boldsymbol{x}_i - \boldsymbol{x}_i'|}$$

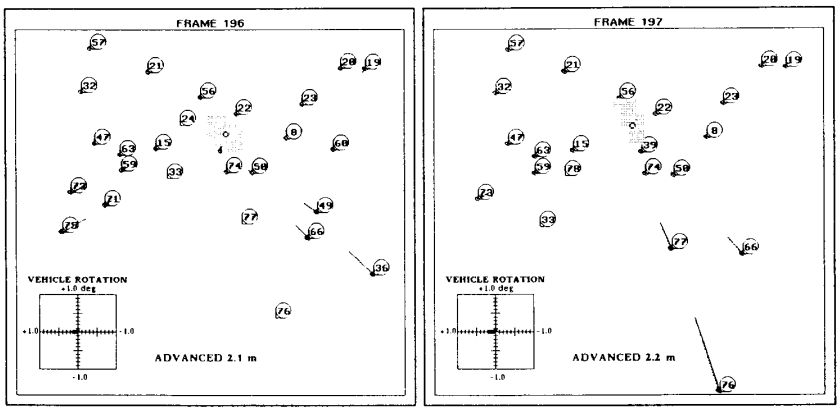

Fig. 18. (*Continued.*)

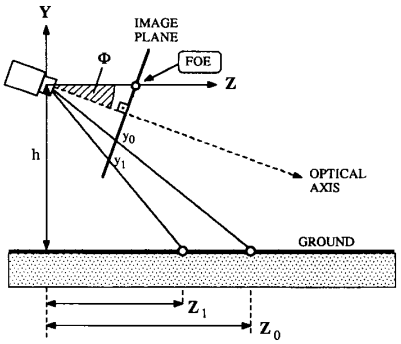

Fig. 19. Side view of the camera traveling parallel to a flat surface. The camera advanced in direction **Z**, such that a 3-D point on the ground moves relative to the camera from $\mathbf{Z}_0$ to $\mathbf{Z}_1$. The depression angle ϕ can be found from the location of the FOE in the image. The height of the camera above the ground is given.

and the final estimate for the vehicle velocity is

$$\tilde{V}(t,t') = \frac{\tilde{\Delta Z}}{t'-t}.$$

This computation was applied to a sequence of real images shown in Section IV.

References

[1] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-7, no. 4, pp. 384–401, 1985.

[2] P. Anandan, "Computing dense displacement fields with confidence measures in scenes containing occlusion," *SPIE Intell. Robots Comput. Vision,* vol. 521, pp. 184–194, 1984.

[3] A. Bandopadhay, B. Chandra, and D. N. Ballard, "Egomotion using active vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1986, pp. 498–503.

[4] S. T. Barnard and W. B. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-2, no. 4, pp. 333–340, July 1980.

[5] B. Bhanu and W. Burger, "DRIVE—Dynamic reasoning from integrated visual evidence," Honeywell Systems & Research Center, Minneapolis, MN, DARPA Rep. DACA 76-86-C-0017, June 1987.

[6] ——, "Qualitative motion detection and tracking of targets from a mobile platform," in *Proc. DARPA Image Understanding Workshop,* Apr. 1988, pp. 289–318.

[7] B. Bhanu, P. Symosek, J. Ming, W. Burger, H. Nasr, and J. Kim, "Qualitative target motion detection and tracking," in *Proc. DARPA Image Understanding Workshop.* Morgan Kaufmann, May 1989.

[8] S. Bharwani, E. Riseman, and A. Hanson, "Refinement of environmental depth maps over multiple frames," in *Proc. IEEE Workshop Motion,* Kiawah Island, SC, May 1986, pp. 73–80.

[9] T. J. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-8, no. 1, pp. 90–99, Jan. 1986.

[10] A. R. Bruss and B. K. P. Horn, "Passive navigation," *Comput. Vision, Graphics, Image Processing,* vol. 21, pp. 3–20, 1983.

[11] W. Burger and B. Bhanu, "Qualitative motion understanding," in *Proc. Tenth Int. Joint Conf. Artificial Intelligence, IJCAI-87,* Milan, Italy, Morgan Kaufmann, Aug. 1987.

[12] ——, "Dynamic scene understanding for autonomous mobile robots," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 1988, pp. 736–741.

[13] ——, "On computing a 'fuzzy' focus of expansion for autonomous navigation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 1989, pp. 563–568.

[14] J. Q. Fang and T. S. Huang, "Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames," *IEEE Trans. Pattern Anal. Machine Intelligence,* vol. PAMI-6, no. 5, pp. 545–554, Sept. 1984.

[15] O. D. Faugeras, F. Lustman, and G. Toscani, "Motion and structure from point and line matches," in *Proc. 1st Int. Conf. Computer Vision,* June 1987, pp. 25–34.

[16] R. M. Haralick, "Using perspective transformations in scene analysis," *Comput. Graphics Image Processing,* vol. 13, pp. 191–221, 1980.

[17] C. Jerian and R. Jain, "Determining motion parameters for scenes with translation and rotation," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-6, no. 4, pp. 523–530, July 1984.

[18] J. Kim and B. Bhanu, "Motion disparity analysis using adaptive windows," Honeywell Systems & Research Center, Tech. Rep. 87SRC38, June 1987.

[19] D. T. Lawton, "Processing translational motion sequences," *Comput. Vision, Graphics, Image Processing,* vol. 22, pp. 114–116, 1983.

[20] D. N. Lee, "The optic flow field: The foundation of vision," *Phil. Trans. Roy. Soc. London B,* vol. 290, pp. 169–179, 1980.

[21] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature,* vol. 293, pp. 133–135, Sept. 1981.

[22] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proc. Roy. Soc. London B,* vol. 208, pp. 385–397, 1980.

[23] H. P. Moravec, "Towards automatic visual obstacle avoidance," in *Proc. 5th Int. Joint Conf. Artificial Intelligence,* Aug. 1977, p. 584.

[24] K. Prazdny, "Determining the instantaneous direction of motion from optical flow generated by a curvilinear moving observer," *Comput. Graphics Image Processing,* vol. 17, pp. 238–248, 1981.

[25] D. Regan, K. Beverly, and M. Cynader, "The visual perception of motion in depth," *Sci. Amer.,* pp. 136–151, July 1979.

[26] J. H. Rieger, "Information in optical flows induced by curved paths of observation," *J. Opt. Soc. Amer.,* vol. 73, no. 3, pp. 339–344, Mar. 1983.
[27] J. H. Rieger and D. T. Lawton, "Processing differential image motion," *J. Opt. Soc. Amer. A,* vol. 2, no. 2, pp. 354–360, Feb. 1985.
[28] J. W. Roach and J. K. Aggarwal, "Determining the movements of objects from a sequence of images," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-2, no. 6, pp. 554–562, 1980.
[29] B. G. Schunck, "Image flow: Fundamentals and future research," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 1985, pp. 560–571.
[30] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-6, no. 1, pp. 13–27, Jan. 1984.
[31] S. Ullman, *The Interpretation of Visual Motion.* Cambridge, MA: MIT Press, 1979.
[32] A. Verri and T. Poggio, "Qualitative information in the optical flow," in *Proc. DARPA Image Understanding Workshop,* Los Angeles, CA, Feb. 1987, pp. 825–834.
[33] J. Weng, T. S. Huang, and N. Ahuja, "Error analysis of motion parameter estimation from image sequences," in *Proc. 1st Int. Conf. Computer Vision,* June 1987, pp. 703–707.

Wilhelm Burger was born in Austria in 1955. He received the undergraduate degree from Johannes Kepler University, Linz, Austria, in 1983 and the M.S. degree in computer science from the University of Utah, Salt Lake City, in 1987. He is currently working toward the Doctoral degree in the area of freeform object recognition and representation.

From 1975 to 1981 he was employed as a system engineer with Kretztechnik Ultrasound, Austria, where he developed imaging hardware for medical ultrasound scanners. During 1986–1987 he was a Research Associate in the Signal and Image Processing group at the Honeywell Systems and Research Center in Minneapolis, MN, where he did work in motion analysis for autonomous vehicle navigation. Since 1987 he has been a faculty member at the Institute of Systems Science at Johannes Kepler University Linz, where he is involved in teaching and research in the area of machine vision. His primary research interests include motion analysis, inexact approaches in vision, freeform object recognition, robust feature extraction, machine learning, and hardware/software environments for image understanding.

Bir Bhanu (S'72–M'82–SM'87) received the B.S. degree (with Honors) in electronics engineering from the Indian Institute of Technology (BHU), Varanasi, India, the M.E. degree (with Distinction) in electronics engineering from Birla Institute of Technology and Science, Pilani, India, the S.M. and E.E. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, the Ph.D. degree in electrical engineering from the Image Processing Institute, University of Southern California, Los Angeles, and the M.B.A. degree from the University of California, Irvine. He also received the Diploma in German from BHU, Varanasi, India.

He is a Honeywell Fellow at Honeywell Systems and Research Center, where he serves as the Principal Investigator of the Scene Dynamics Program from DARPA, the Obstacle Detection Program from NASA, and the Machine Learning Program from a government agency. Additionally, he is conducting sponsored and IR&D research efforts on machine learning, robotic vehicle navigation, target recognition and tracking, object modeling, contextual analysis, multisensor integration, parallel algorithms, scientific performance evaluation of image understanding algorithms/systems, and photointerpretation and surveillance. He has also worked with IBM on image processing, INRIA-France on 3-D object recognition and Ford Aerospace and Communications Corporation on automatic target recognition. While on the faculty of the Department of Computer Science, University of Utah, he was the Principal Investigator on several NSF and industry funded research projects in machine intelligence. His current interests are computer vision, robotics, machine learning, neural networks and neurocomputers, object modeling, multisensor integration, distributed sensing and control, parallel computer architectures, and applications of artificial intelligence.

Dr. Bhanu has over 100 reviewed technical publications and 5 patents in the areas of his interest. He has given national short courses on intelligent automatic target recognition. He is a reviewer to over two dozen technical publications and government agencies. He was the Guest Editor of a special issue of IEEE *Computer* on "CAD-Based Robot Vision" published in August 1987. He is listed in the *American Men and Women of Science, Who's Who in the West, Personalities of Americas* and *5000 Personalities of the World.* He is a member of ACM, AAAI, Sigma Xi, Pattern Recognition Society, SPIE, and the IEEE Computer Society.