Qualitative Motion Detection and Tracking of Targets from a Mobile Platform

Bir Bhanu and Wilhelm Burger

Honeywell Systems and Research Center
3660 Technology Drive, Minneapolis, MN 55418

ABSTRACT

The problem of understanding scene dynamics is to find consistent and plausible 3-D interpretations for any change observed in the 2-D image sequence. Due to the motion of the Autonomous Land Vehicle (ALV), stationary objects in the scene generally do not appear stationary in the image, whereas moving objects are not necessarily seen in motion. The three main tasks of our novel approach for target motion detection and tracking are: (a) to estimate the vehicle's motion, (b) to derive the 3-D structure of the stationary environment, and (c) to detect and classify the motion of individual targets in the scene. These three tasks strongly depend on each other. The direction of heading (i.e. translation) and rotation of the vehicle are estimated with respect to stationary locations in the scene. The focus of expansion (FOE) is not determined as a particular image location, but as a region of possible FOE-locations called the *Fuzzy FOE*. We present a qualitative strategy of reasoning and modeling for the perception of 3D space from motion information. Instead of refining a single quantitative description of the observed environment over time, multiple *qualitative interpretations* are maintained simultaneously. This offers superior robustness and flexibility over traditional numerical techniques which are often ill-conditioned and noise-sensitive. A rule-based implementation of this approach is discussed and results on real ALV imagery are presented.

1. INTRODUCTION

Visual information is an indispensable clue for the successful operation of an Autonomous Land Vehicle (ALV). Even with the use of sophisticated inertial navigation systems, the accumulation of position errors requires periodic corrections. Operation in unknown environments or mission tasks involving search, rescue, or manipulation critically depend upon visual feedback.

Assessment of scene dynamics becomes vital when moving objects may be encountered, e.g., when the ALV follows a convoy, approaches other vehicles, or has to detect moving threats. For the given case of a moving camera, image motion can also supply important information about the spatial layout of the environment ("motion stereo") and the actual movements of the ALV. This is a valuable input for navigation and vehicle control, i.e., steering, accelerating, and braking.

Previous work in motion analysis has mainly concentrated on numerical approaches for the recovery of 3D motion and scene structure from 2D image sequences. Recently, Nagel[11] gave an excellent review. The most common approach is to estimate 3D structure and motion in one computational step by solving a system of linear or non-linear equations.[5,17] This technique is characterized by several severe limitations.

First, it is known for its notorious noise-sensitivity. To overcome this problem, some researchers have extended this technique to cover multiple frames.[3,7] Secondly, it is designed to analyze the relative motion and 3D structure of a single rigid object. To estimate the ALV's egomotion and the scene structure, the environment would have to be treated as a large rigid object. However, rigidness of the environment cannot be guaranteed due to the possible presence of moving objects in the scene. But what is the consequence of accidentally including a moving 3D point into the system of equations? In the best case, the solution (in terms of motion and structure) would exhibit a large residual error, indicating some non-rigid behavior. The point in motion, however, cannot be immediately identified from this solution alone. In the worst case (for some forms of motion), the system may converge towards a rigid solution (with small error) in spite of the actual movement in the point set. This again shows another (third) limitation: there is no suitable means of expressing the ambiguity and uncertainty inherent to dynamic scene analysis.

The approach that we propose is novel in two important aspects. First, scene structure is not treated as a mere by-product of the motion computation but as a valuable means to overcome some of the ambiguities of dynamic scene analysis. The key idea is to use the description of the scene's 3D structure as a link between motion analysis and other processes that deal with spatial perception, such as shape-from-occlusion, stereo, spatial reasoning, etc. A 3D interpretation of a moving scene can only be correct if it is acceptable by all the processes involved.

Secondly, numerical techniques have been largely replaced by a qualitative strategy of reasoning and modeling. The use of qualitative techniques in computer vision has been of growing interest recently.[16,19] Basically, instead of having a system of equations approach a single rigid (but possibly incorrect) numerical solution, we maintain multiple qualitative interpretations of the scene. All the existing interpretations are kept consistent with the observations made in the past. The main advantage of this approach is that a new interpretation can be supplied immediately when the currently favored interpretation turns out to be unplausible.

These interpretations are built in three separate steps (see Figure 1). First, significant features (points, boundaries, corners, etc.) are extracted from the image and the 2D dis-

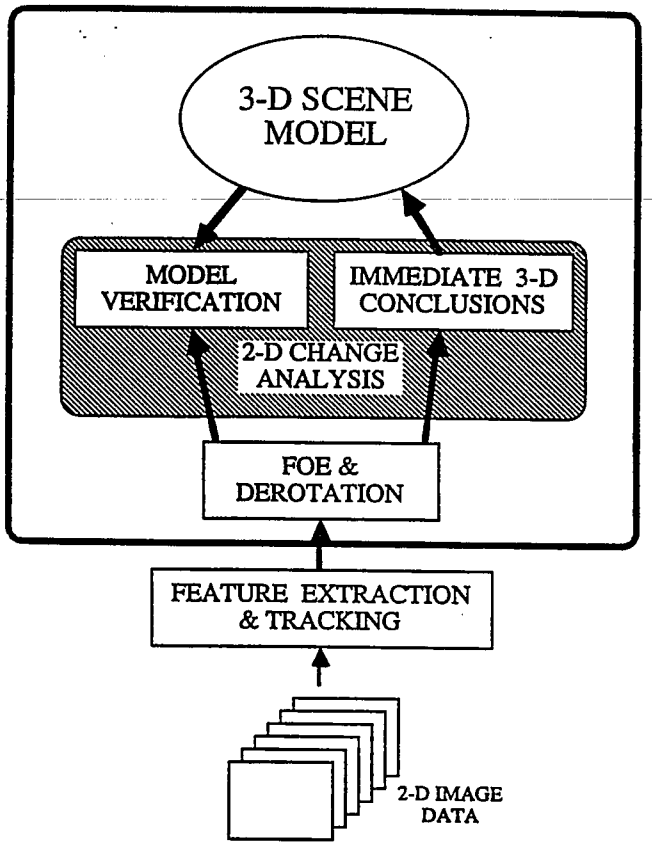


Figure 1. Main steps of the Qualitative Motion Detection and Tracking approach.

placement vectors are computed for this set of features. For the examples shown here, points were selected and tracked between individual frames. Automatic techniques suitable for this task can be found elsewhere.[2, 10] In the second step, the vehicle's direction of translation, i.e. the Focus of Expansion (FOE), and the amount of rotation in space are determined. The effects of vehicle motion on the FOE computation is described in section 2. Almost all the necessary numerical computation is performed in the FOE computation stage, which is described in section 3. The third step (2D Change Analysis) constructs an internal 3D model of the scene. Section 4 outlines the concepts and operation of this Qualitative Scene Model. Experiments with our approach on real imagery taken from the moving ALV are discussed in section 5. Finally, section 6 presents the conclusions of the qualitative motion detection and tracking system.

2. EFFECTS OF VEHICLE MOTION

The first step of our approach is to estimate the vehicle's motion relative to the stationary environment using visual information. Arbitrary movement of an object in 3D space and thus the movement of the vehicle itself can be described as a combination of translation and rotation. While knowledge about the composite vehicle motion is essential for control purposes, only translation can supply information about the spatial layout of the 3D scene (motion stereo). This, however, requires the removal of all image effects resulting from vehicle rotation. For this purpose, we discuss the changes upon the image that are caused by individual application of the "pure" motion components.

2.1 Viewing Geometry

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[13, 15] have used a velocity-based formulation of the problem, the following treatment views motion in discrete time steps.

Given the world coordinate system (X Y Z) shown in Figure 2, a translation $\mathbf{T} = (U\ V\ W)^T$ applied to a point in 3D $\mathbf{X} = (X\ Y\ Z)^T$ is accomplished through vector addition:

$$\mathbf{X}' = \mathbf{T} + \mathbf{X} = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{1}$$

A 3D rotation $\mathbf{R}$ about an arbitrary axis through the origin of the coordinate system can be described by successive rotations about its three axes:

$$\mathbf{R} = \mathbf{R}_\phi\, \mathbf{R}_\theta\, \mathbf{R}_\psi \tag{2}$$

where

$$\mathbf{R}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \text{ rotation about the X–axis,} \tag{3a}$$

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \text{ rotation about the Y–axis,} \tag{3b}$$

$$\mathbf{R}_\psi = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ rotation about the Z–axis.} \tag{3c}$$

A general rigid motion in space consisting of translation and rotation is described by the transformation

$$\mathbf{M}: \mathbf{X} \rightarrow \mathbf{X}' = \mathbf{R}_\phi\, \mathbf{R}_\theta\, \mathbf{R}_\psi\, (\mathbf{T}+\mathbf{X}) \tag{4}$$

Its six degrees of freedom are U, V, W, ϕ, θ and ψ.

This decomposition is not unique because the translation could be as well applied after the rotation. 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.

To model the movements 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 is located in the lens center of the camera.

2.2 Image Effects of 3D Camera Motion

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 camera image.

Under perspective imaging, a point in space $\mathbf{X} = (X\ Y\ Z)^T$ is projected onto a location on the image plane $\mathbf{x} = (x\ y)^T$ such that

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z}, \tag{5}$$

where f is the focal length of the camera (see Figure 2).

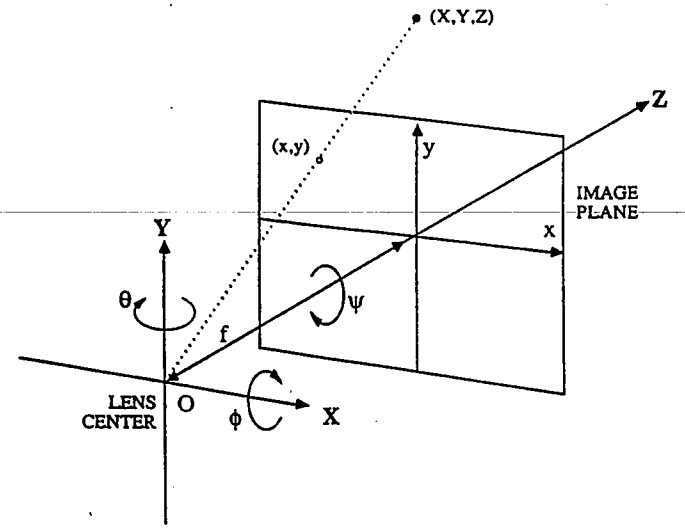


Figure 2: Camera Model showing the coordinate system, lens center, image plane and angles of rotation. The origin of the coordinate system is located at the lens center. The focal length f is the distance between the lens center and the image plane.

2.2.1 Effects of Pure Camera Rotation

When the camera is rotated around its lens center, the acquired image changes but no new views of the environment are obtained. Camera rotation merely maps the image into itself.

The most intuitive effect results from pure rotation about the Z-axis of the camera-centered coordinate system, which is also the optical axis. Any point in the image moves along a circle centered at the image location $\mathbf{x} = (0\ 0)$.

In practice, however, the amount of rotation of the vehicle about the Z-axis is small. Therefore, vehicle rotation is confined to the X- and Y-axis, where significant amounts of rotation occur.

The vehicle undergoing rotation about the X-axis by an angle $-\phi$ and the Y-axis by an angle $-\theta$ moves each 3D point $\mathbf{X}$ to point $\mathbf{X}'$ relative to the camera.

$$\mathbf{X} \rightarrow \mathbf{X}' = \mathbf{R}_\phi \cdot \mathbf{R}_\theta \cdot \mathbf{X} \tag{6}$$

$$= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ \sin\phi\sin\theta & \cos\phi & -\sin\phi\cos\theta \\ -\cos\phi\sin\theta & \sin\phi & \cos\phi\cos\theta \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

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

$$x' = f\,\frac{X\cos\theta + Z\sin\theta}{-X\cos\phi\sin\theta + Y\sin\phi + Z\cos\phi\cos\theta} \tag{7a}$$

$$y' = f\,\frac{X\sin\phi\sin\theta + Y\cos\phi - Z\sin\phi\cos\theta}{-X\cos\phi\sin\theta + Y\sin\phi + Z\cos\phi\cos\theta} \tag{7b}$$

Inverting the perspective transformation for the original image point $\mathbf{x}$ yields

$$X = \frac{1}{f}\,Z\,x \quad \text{and} \quad Y = \frac{1}{f}\,Z\,y\,. \tag{8}$$

The 2D *rotation mapping* $\mathbf{r}_\phi\mathbf{r}_\theta$ which moves each image point $\mathbf{x} = (x\ y)$ into the corresponding image point $\mathbf{x}' = (x'y')$ under camera rotation $\mathbf{R}_\phi\mathbf{R}_\theta$ (i.e., a particular sequence of *pan* and *tilt*) is given by

$$\mathbf{R}_\phi\mathbf{R}_\theta\ (\mathbf{X}) : \mathbf{X} \rightarrow \mathbf{X}'$$

$$\mathbf{r}_\phi\mathbf{r}_\theta\ (\mathbf{x}) : \mathbf{x} = (x\ y) \rightarrow \mathbf{x}' = (x'\ y')$$

$$x' = f\,\frac{x\cos\theta + f\sin\theta}{-x\cos\phi\sin\theta + y\sin\phi + f\cos\phi\cos\theta} \tag{9a}$$

$$y' = f\,\frac{x\sin\phi\sin\theta + y\cos\phi - f\sin\phi\cos\theta}{-x\cos\phi\sin\theta + y\sin\phi + f\cos\phi\cos\theta}\,. \tag{9b}$$

It is important to notice that this transformation contains no 3D variables and is therefore a mapping of the image onto itself. This demonstrates that no additional information about the 3D structure of the scene can be obtained under pure camera rotation.

An interesting property of this mapping should be mentioned at this point, which might not be obvious. Moving an image point on a diagonal passing through the center of the image at 45° by only rotating the camera does *not* result in equal amounts of rotation about the X- and the Y-axis. This is again a consequence of the successive application of the two rotations $\mathbf{R}_\theta$ and $\mathbf{R}_\phi$, since the first rotation about the Y-axis also changes the orientation of the camera's X-axis in 3D space. It also explains why the pair of equations in (7) is not symmetric with respect to θ and ϕ.

2.2.2 Measuring the Amount of Camera Rotation

The problem to be solved is the following: Given are two image locations $\mathbf{x}_0$ and $\mathbf{x}_1$, which are the observations of the same 3D point at time t_0 and time t_1. What is the amount of rotation $\mathbf{R}_\phi$ and $\mathbf{R}_\theta$ which applied to the camera between time t_0 and time t_1, would move image point $\mathbf{x}_0$ onto $\mathbf{x}_1$ assuming that no camera translation occurred at the same time?

If $\mathbf{R}_\phi$ and $\mathbf{R}_\theta$ are applied to the camera separately, the points in the image move along hyperbolic paths.[12] If pure *horizontal* rotation were applied to the camera, a given image point $\mathbf{x}_0$ would move on a path described by

$$\mathbf{r}_\theta\ (\mathbf{x}_0)\text{: } y^2 = y_0^2\,\frac{f^2 + x^2}{f^2 + x_0^2}\,. \tag{10}$$

Similarly pure *vertical* camera rotation would move an image point $\mathbf{x}_1$ along

$$\mathbf{r}_\phi\ (\mathbf{x}_1)\text{: } x^2 = x_1^2\,\frac{f^2 + y^2}{f^2 + y_1^2}\,. \tag{11}$$

Since the 3D rotation of the camera is modeled as being performed in two separate steps ($\mathbf{R}_\theta$ followed by $\mathbf{R}_\phi$), the *rotation mapping* $\mathbf{r}_\phi\mathbf{r}_\theta$ can also be separated into $\mathbf{r}_\theta$ followed by $\mathbf{r}_\phi$. In the first step, applying pure (horizontal) rotation around the Y-axis $\mathbf{r}_\theta$, point $\mathbf{x}_0$ is moved to an intermediate image location $\mathbf{x}_c$. The second step, applying pure (vertical) rotation around the X-axis $\mathbf{r}_\phi$, takes point $\mathbf{x}_c$ to the final image location $\mathbf{x}_1$. This can be expressed as

$$\mathbf{r}_{\phi\theta} = \mathbf{r}_\phi\ \mathbf{r}_\theta \quad \text{,where} \tag{12}$$

$$\mathbf{r}_\theta\text{: } \mathbf{x}_0 = (x_0\ y_0) \rightarrow \mathbf{x}_c = (x_c\ y_c)$$

$$\mathbf{r}_\phi\text{: } \mathbf{x}_c = (x_c\ y_c) \rightarrow \mathbf{x}_1 = (x_1\ y_1)\,.$$

As shown in Figure 3, the image point $\mathbf{x}_c = (x_c\ y_c)$ is the intersection point of the hyperbola passing through $\mathbf{x}_0$ result-

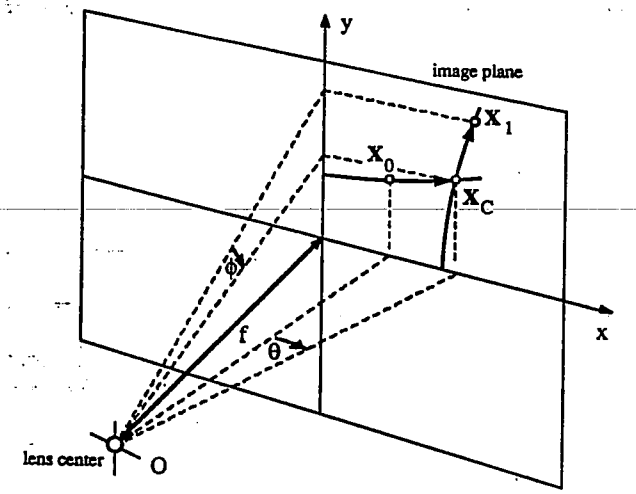


Figure 3: Successive Application of Horizontal and Vertical Rotation. The image point x_0 is to be moved to location x_1 by pure horizontal and vertical camera rotation. Horizontal rotation (about the Y-axis) is applied first, moving x_0 to x_C, which is the intersection point of the two hyperbolic paths for horizontal and vertical rotation. In a second step x_c is taken to x_1. Then the two rotation angles θ and φ are found directly.

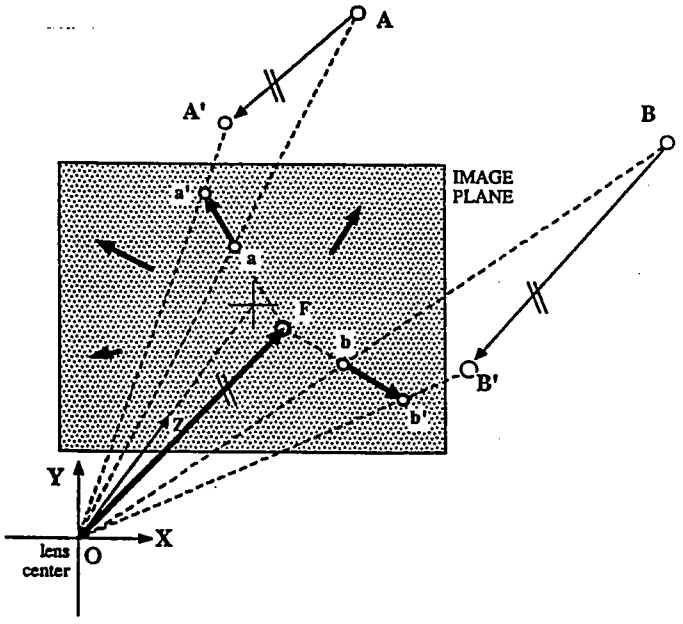


Figure 4: Location of the focus of expansion (FOE). With pure vehicle translation, points in the environment (A, B) move along 3D vectors parallel to the vector pointing from the lens center to the FOE in the camera plane. These vectors form parallel lines in space which have a common vanishing point (the FOE) in the perspective image.

ing from horizontal camera rotation (10) with the hyperbola passing through x_1 resulting from vertical camera rotation (11). Intersecting the two hyperbolae gives the image point x_c, with

$$x_c = f\,x_1 \left[\frac{f^2 + x_0^2 + y_0^2}{(f^2 + x_0^2)(f^2 + y_1^2) - x_1^2\,y_0^2} \right]^{1/2} \qquad (13a)$$

$$y_c = f\,y_0 \left[\frac{f^2 + x_1^2 + y_1^2}{(f^2 + x_0^2)(f^2 + y_1^2) - x_1^2\,y_0^2} \right]^{1/2} . \qquad (13b)$$

The amount of camera rotation necessary to map x_0 onto x_1 by applying R_θ followed by R_ϕ is finally obtained as

$$\theta = \tan^{-1}\frac{x_c}{f} - \tan^{-1}\frac{x_0}{f} \qquad (14)$$

$$\phi = \tan^{-1}\frac{y_c}{f} - \tan^{-1}\frac{y_1}{f} . \qquad (15)$$

2.2.3 Effects of Pure Camera Translation

When the vehicle undergoes pure translation between time t and time t', every point on the vehicle is moved by the same 3D vector $\mathbf{T} = (U\ V\ W)^T$. Again, the same effect is achieved by keeping the camera fixed and moving every point X_i in the environment to X_i' by applying $-\mathbf{T}$.

Since every stationary point in the environment undergoes the same translation relative to the camera, the imaginary lines between corresponding points $\overline{X_i X_i'}$ are parallel in 3D space.

It is a fundamental result from perspective geometry[4] that the images of parallel lines pass through a single point in the image plane called a *vanishing point*. When the camera moves along a straight line, every (stationary) image point seems to expand from this vanishing point or contract towards it when the camera moves backwards. This particular image location is therefore commonly referred to as the *Focus of Expansion* (FOE) or the *Focus of Contraction* (FOC). Each displacement vectors passes through the FOE creating the typical radial expansion pattern shown in Figure 4.

As can be seen in Figure 4, the straight line passing through the lens center of the camera and the FOE is also parallel to the 3D displacement vectors. Therefore, the 3D vector $\vec{OF}$ points in the direction of camera translation in space. Knowing the internal geometry of the camera (i.e., the focal length), the direction of vehicle translation can be determined by locating the FOE in the image. The actual translation vector T applied to the camera is a multiple of the vector $\vec{OF}$ which supplies only the *direction* of camera translation but not its *magnitude*. Therefore,

$$\mathbf{T} = \lambda\,\vec{OF} = \lambda\,[\,x_f\ y_f\ f\,]^T \quad , \lambda \in R\,. \qquad (16)$$

Since most previous work incorporated a velocity-based model of 3D motion, the *Focus of Expansion* has commonly been interpreted as the *direction of instantaneous heading*, i.e., the direction of vehicle translation during an infinitely short period in time. When images are given as "snapshots" taken at discrete instances of time, the movements of the vehicle must be modeled accordingly as discrete movements from one position in space to the next.

Therefore, the FOE cannot be interpreted as the momen-

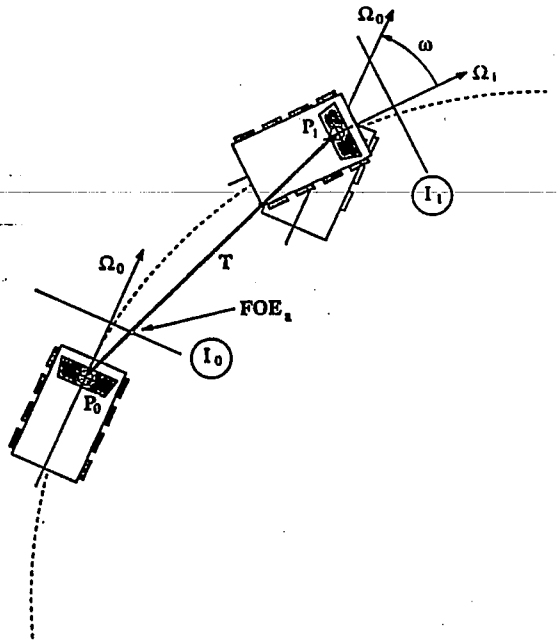


Figure 5: Concept of the FOE for discrete time steps. The vehicle's motion between two points in time can be decomposed into a translation followed by a rotation: the image effects of pure translation (FOE_a) are observed in image I_0. This scheme is used throughout this work.

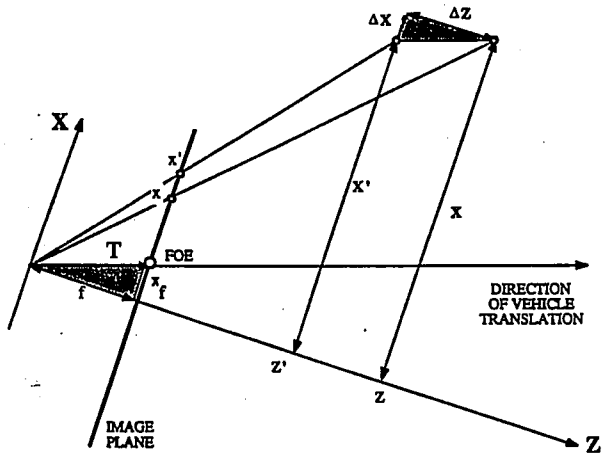


Figure 6: Amount of expansion from the FOE for discrete time steps. The camera moves by a vector T in 3D space, which passes through the lens center and the FOE in the camera plane. The 3D Z-axis is also the optical axis of the camera.

tary direction of translation at a certain point in time, but rather as the direction of *accumulated* vehicle translation over a *period* of time.

Figure 5 shows the top view of a vehicle traveling along a curved path at two instances in time t_0 and t_1. The position of the vehicle in space is given by the position of a reference point on the vehicle P and the orientation of the vehicle Ω. Figure 5 also displays the adopted scheme of 3D motion decomposition: First the translation T is applied which shifts the vehicle's reference point (i.e., the lens center of the camera) from position P_0 to position P_1 without changing the vehicle's orientation Ω_0. The 3D translation vector intersects the image plane at FOE_a. In the second step the vehicle is rotated by ω to the new orientation Ω_1. Translation T transforms image I_0 into image I_1', which again is transformed into I_1 by rotation ω. The important fact is that FOE_a is observed at the transition from image I_0 to image I_1', which is obtained by *derotating* image I_1 by $-\omega$. Throughout the rest of this work, this scheme (Figure 5) is used as a model for vehicle motion.

2.2.4 Measuring the Amount of Camera Translation

Figure 6 shows the geometric relationships for the 2D 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 $\mathbf{T} = (x_f \; f)^T$.

A stationary 3D point is observed at two instances of time, which moves in space relative to the camera from X to X', resulting in two images x and x'.

$$\mathbf{X} = \begin{bmatrix} X \\ Z \end{bmatrix} \quad \text{and} \quad \mathbf{X}' = \begin{bmatrix} X' \\ Z \end{bmatrix} = \begin{bmatrix} X - \Delta X \\ Z - \Delta Z \end{bmatrix} . \tag{17}$$

Using the inverse perspective (8) transformation yields

$$Z = \frac{f}{x} X \quad \text{and} \tag{18}$$

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

From similar triangles (shaded in Figure 6)

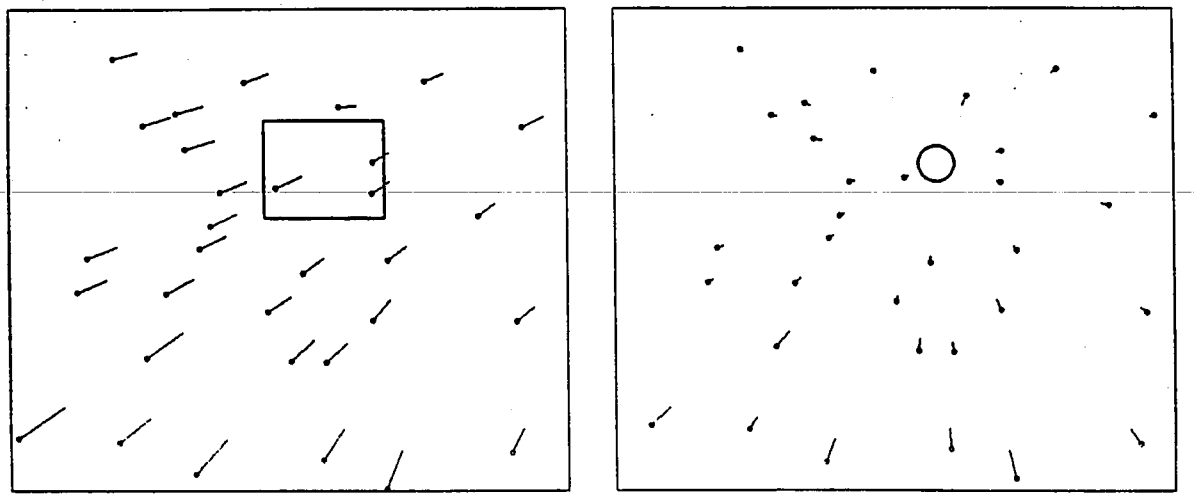
$$\frac{\Delta X}{x_f} = \frac{\Delta Z}{f} , \tag{19}$$

and therefore

$$Z = \Delta Z \, \frac{x' - x_f}{x' - x} = \Delta Z \left(1 + \frac{x - x_f}{x' - x} \right) . \tag{20}$$

Thus, the rate of expansion of image points from the FOE contains direct information about the distance of the corresponding 3D points from the camera. Consequently, if the vehicle is moving along a straight line and the FOE has been located, the 3D structure of the scene can be determined from the expansion pattern in the image. However, the distance Z of a 3D 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 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



(a) *(b)*

Figure 7: Simulated displacement field caused by a combination of horizontal and vertical rotation and vehicle translation. *(a)* The rectangle marks the area of search for the FOE. *(b)* The *derotated* displacement with the FOE marked by a circle.

- the FOE can be located in a small area, and
- the observed image points exhibit significant expansion away from the FOE.

As will be shown in the following section, imaging 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 3D points $\{(\mathbf{X}_i\, \mathbf{X}_i')\}$ is observed, then of course the translation in the Z-direction is the same for every point.

$$Z_i - Z_i' = Z_j - Z_j' = \Delta Z \quad \text{for all } i,j\,. \tag{21}$$

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

$$Z_i = \propto \frac{x_i' - x_f}{x_i' - x_i}\,, \tag{22}$$

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

The effects of camera translation **T** can be formulated as a mapping **t** of a set of image locations $\{\mathbf{x}_i\}$ into another set of image locations $\{\mathbf{x}_i'\}$. Unlike in the case of pure camera rotation, this mapping not only depends upon the 3D translation vector but also upon the actual 3D location of each individual point observed. Therefore, in general, **t** is *not* simply a mapping of the image onto itself.

However, one important property of **t** can be described exclusively in the image plane, namely that each point must map onto a straight line passing through the original point and one unique location in the image (the FOE). This means that if the vehicle is undergoing pure translation, then there must exist an image location $\mathbf{x}_f$ such that the mapping **t** satisfies the condition

radial–mapping $\mathbf{t}\ (\mathbf{x}_f, \mathbf{I}, \mathbf{I}')$:

$$\mathbf{t} = \{\ (\mathbf{x}_i\,,\, \mathbf{x}_i') \in \mathbf{I}\times\mathbf{I}' \mid \mathbf{x}_i' = \mathbf{x}_i + \mu_i\,(\mathbf{x}_i - \mathbf{x}_f)\,, \tag{23}$$

$$\mu_i \in R,\ \mu_i \geq 0\ \}\,.$$

2.2.5 Combined Effects of Translation and Rotation

When the vehicle is not undergoing pure translation or rotation but combined 3D motion of the form $\mathbf{R}_\phi\, \mathbf{R}_\theta\, \mathbf{T}$, the effects in the image are described by a transformation **d** (for *displacement*) which is a combination of $\mathbf{r}_\phi$, $\mathbf{r}_\theta$ and **t**:

$$\mathbf{d} : \mathbf{I} \rightarrow \mathbf{I}' = \mathbf{r}_\phi\, \mathbf{r}_\theta\, \mathbf{t}\ (\mathbf{I})\ , \tag{24}$$

where $\mathbf{I} = \{\ \mathbf{x}_i\ \}$, $\mathbf{I}' = \{\ \mathbf{x}_i'\ \}$ are the two sets of corresponding image points.

Figure 7 shows a typical displacement field for a camera undergoing horizontal and vertical rotation as well as translation. The points $\mathbf{x}_i \in \mathbf{I}$ are marked with small circles.

By decomposing a composite displacement field **d** into its three components $\mathbf{r}_\phi$, $\mathbf{r}_\theta$, and **t**, the vehicle's rotation and direction of translation in space can be computed from the information available in the image. This problem is addressed in the following section.

3. DECOMPOSITION OF IMAGE MOTION

3.1 Problem Statement

As discussed in the previous section, the 3D motion **M** of the vehicle is modeled by a translation **T** followed by a rotation $\mathbf{R}_\theta$ about the Y-axis and a rotation $\mathbf{R}_\phi$ about the X-axis:

$$\mathbf{M} = \mathbf{R}_\phi\, \mathbf{R}_\theta\, \mathbf{T}\ . \tag{25}$$

This results in a mapping $\mathbf{d}$ from the original image $\mathbf{I}_0$ at time t_0 into the new image $\mathbf{I}_1$ at time t_1.

$$\mathbf{d} : \mathbf{I}_0 \rightarrow \mathbf{I}_1 = \mathbf{r}_\phi \, \mathbf{r}_\theta \, \mathbf{t} \, \mathbf{I}_0 = \mathbf{r}_\phi \, \mathbf{r}_\theta \, \mathbf{I}_0' . \qquad (26)$$

The intermediate image $\mathbf{I}_0'$ in (26) is the result of the translation component of the vehicle's motion and has the property of being a radial mapping (23). Unlike the two images $\mathbf{I}_0$ and $\mathbf{I}_1$, which are actually given, the image $\mathbf{I}_0'$ is generally not observed, except when the camera rotation is zero. It serves as an intermediate result to be reached during the separation of translational and rotational motion components.

The question at this point is whether there exists more than one combination of rotation mappings $\mathbf{r}_\phi$ and $\mathbf{r}_\theta$ which would satisfy this requirement, i.e., if the solution is *unique*. It has been pointed out in the previous section that the decomposition of 3D motion into $\mathbf{R}_\phi$, $\mathbf{R}_\theta$, $\mathbf{R}_\Psi$, and $\mathbf{T}$ is unique for a fixed order of application. This does not imply, however, that the effects of 3D motion upon the perspective image are unique as well.

Tsai and Huang[17] have shown that seven points in two perspective views suffice to obtain a unique interpretation in terms of rigid body motion and structure, except for a few cases where points are arranged in some very special configuration in space. Ullman[18] reports computer experiments which suggest that six points are sufficient in many cases and seven or eight points yield unique interpretations in most cases.

Due to its design and the application, however, the motion of the ALV in space is quite restricted. The vehicle can only travel upright on a surface and its large wheelbase allows for only relatively small changes in orientation. It is also heavy and thus exhibits considerable inertia. Therefore, the final motion parameters must lie within a certain narrow range and it can be expected that a unique solution can be found even in cases when the number of points is near the above minimum.

The fact that

$$\mathbf{I}_0' = \mathbf{r}_\theta^{-1} \, \mathbf{r}_\phi^{-1} \, \mathbf{I}_1 = \mathbf{t} \, \mathbf{I}_0 \qquad (27)$$

suggests two different strategies for separating the motion components:

(1) *FOE from Rotation*: Successively apply combinations of inverse rotation mappings $\mathbf{r}_{\theta_1}^{-1} \, \mathbf{r}_{\phi_1}^{-1}$, $\mathbf{r}_{\theta_2}^{-1} \, \mathbf{r}_{\phi_2}^{-1}$,... $\mathbf{r}_{\theta_k}^{-1} \, \mathbf{r}_{\phi_k}^{-1}$ to the second image $\mathbf{I}_1$, until the resulting image $\mathbf{I}'$ is a radial mapping with respect to the original image $\mathbf{I}_0$. Then locate the FOE $\mathbf{x}_{f_k}$ in $\mathbf{I}_0$.

(2) *Rotation from FOE*: Successively select FOE-locations (different directions of vehicle translation) $\mathbf{X}_{f_1}$, $\mathbf{X}_{f_2}$,... $\mathbf{X}_{f_i}$ in the original image $\mathbf{I}_0$ and then determine the inverse rotation mapping $\mathbf{r}_{\theta_i}^{-1} \, \mathbf{r}_{\phi_i}^{-1}$ that yields a radial mapping with respect to the given FOE $\mathbf{x}_{f_i}$ in the original image $\mathbf{I}_0$.

Both alternatives were investigated under the assumption of restricted, but realistic vehicle motion, as stated earlier. It turned out that the major problem in the *FOE-from-Rotation* approach is to determine if a mapping of image points is (or is close to being) radial when the location of the FOE is unknown. Of course, 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.

Although there have been a number of suggestions for FOE-algorithms in the past,[8,12,15] no results of implementations have been demonstrated on real outdoor imagery. One reason for the absence of useful results might be that most researchers have tried to locate the FOE in terms of a single, distinct image location. In practice, however, the noise generated by merely digitizing a perfect translation displacement field may keep the resulting vectors from passing 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[14] to be as large as 10° and up to 20° in the presence of large rotations.

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.

3.2 FOE from Rotation

In this method, the image motion is decomposed in two steps. First, the rotational components are estimated and their inverses are applied to the image, thus partially "derotating" the image. If the rotation estimate was accurate, the resulting displacement field after derotation would diverge from a single image location (the FOE). The second step verifies that the displacement field is actually radial and determines the location of the FOE. For this purpose, 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.

3.2.1 Estimating the rotational components

Each vector in the displacement field is the sum of vector components caused by camera rotation and camera translation. Since the displacement caused by translation depends on the depth of the corresponding points in 3D space (equation 18), points located at a large distance from the camera are not significantly affected by camera translation. Therefore, one way of estimating vehicle rotation is to compute θ and ϕ from displacement vectors which are *known* to belong to points at far distance. Under the assumption that those displacement vectors are only caused by rotation, equations 14 and 15 can be applied to find the two angles. In some situations, distant points are selected easily. For example, points on the horizon are often located at a sufficient distance from the vehicle. Image points close to the axis of translation would be preferred because they expand from the FOE slower than other points at the same depth.

However, points at far distances may not always be available or may not be *known* to exist in the image. In those cases, the following method for estimating the rotational components can be used. The design of the ALV (and most other mobile robots) does not allow rapid changes in the direction of vehicle heading. Therefore, it can be assumed that the motion of the camera between two frames is constrained, such that the FOE can change its location only within a certain range. If the FOE was located in one frame, the FOE in the subsequent frame must lie in a certain image region around the previous FOE location.

Figure 8(a) illustrates this situation. The FOE of the previous frame was located at the center of the square, thus the FOE in the given frame must be inside this square.

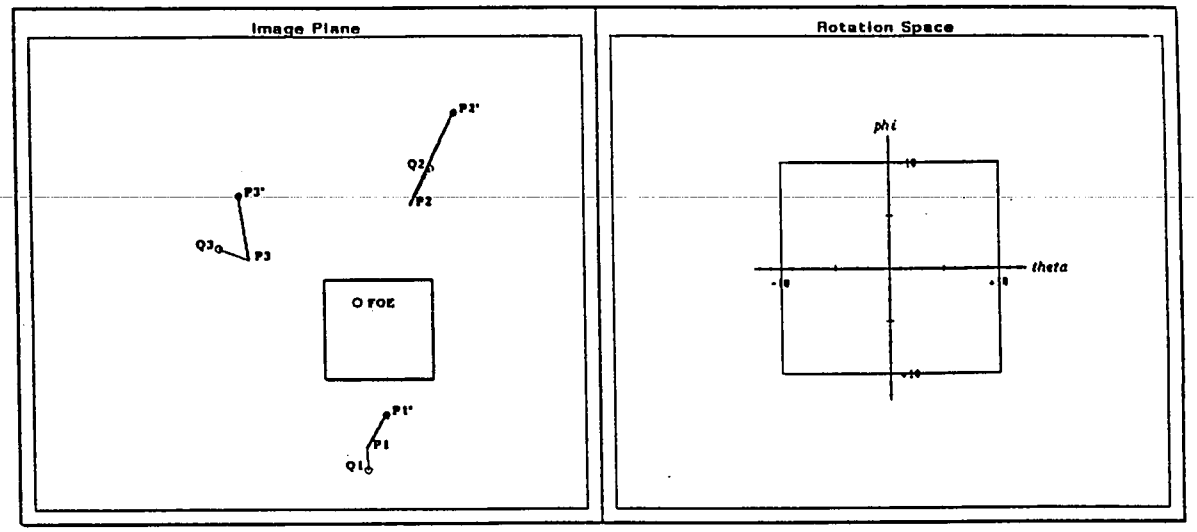


Figure 8: Image Plane and Rotation Space. The displacement field in the Image Plane *(a)* contains three vectors (P1→P1′, P2→P2′, P3→P3′). The previous FOE was observed at the center of the square, which outlines the region of search for the current FOE. The translational displacement components (P1→Q1, P2→Q2, P3→Q3) and the current location of the FOE are unknown but marked in this picture. The initial range of possible camera rotations is ±10° in either direction, indicated by a square in Rotation Space *(b)*.

Three displacement vectors are shown (P1→P1′, P2→P2′, P3→P3′). The translational components (P1→Q1, P2→Q2, P3→Q3) of those displacement vectors and the FOE (inside the square) are not known at this point in time.

The main idea of this technique is to determine the possible range of camera rotations which would be consistent with the FOE lying inside the marked region. Since the camera rotates about two axes, the resulting range of rotations can be described as a *region* in a 2D space. Figure 8(b) shows this *Rotation Space* with the two axes *theta* and *phi* corresponding to the amount of camera rotation around the Y-axis and the X-axis respectively. The initial rotation estimate is a range of ±10° in both directions which is indicated by a square in rotation space.

In general, the range of possible rotations is described by a closed, convex polygon in rotation space. A particular rotation (θ′,φ′) is possible if its application to every displacement vector (i.e., to its endpoint) yields a new vector which lies on a straight line passing through the maximal FOE-region. The region of possible rotations is successively constrained by applying the following steps for every displacement vector (Figure 9):

(a) Apply the rotation mapping defined by the *vertices* of the rotation polygon to the endpoint $\mathbf{P}'$ of the displacement vector. This yields a set of image points $\bar{\mathbf{P}}_i$.

(b) Connect the points $\bar{\mathbf{P}}_i$ to a closed polygon in the image. This polygon is similar to the rotation polygon but distorted by the nonlinear rotation mapping (Figure 9(a)).

(c) Intersect the polygon in the image with the open triangle formed by the starting point P of the displacement vector and the two tangents onto the FOE-region. The result is a new (possibly empty) polygon in the image plane.

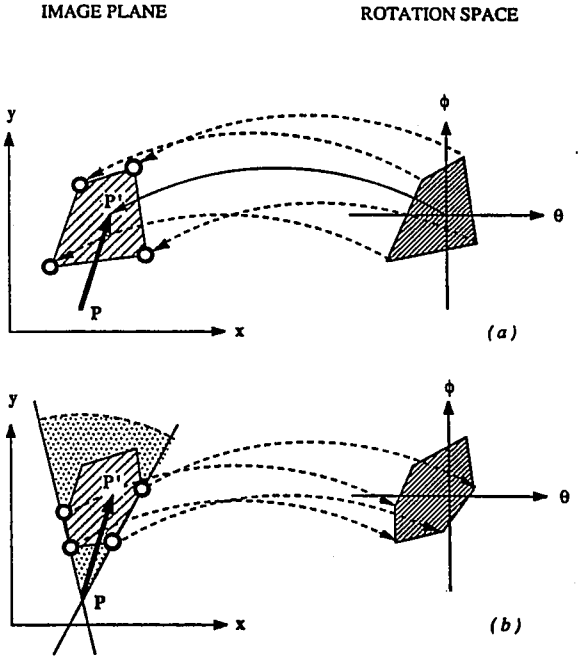


Figure 9: Successively constraining the range of possible camera rotations. *(a)* The rotation mappings corresponding to the vertices of the current rotation polygon are applied to every displacement vector (P→P′). This yields a similar but distorted polygon in the image plane. *(b)* The polygon in the image is intersected with the open triangle defined by the tangents to the maximal FOE-region. Rotations that would bring the endpoint of the displacement vector outside this triangle are not feasible. The new vertices on the polygon are mapped back into rotation space.

(d) Map the new polygon from the image plane back into the rotation space (Figure 9(b)).

(e) If the rotation polygon is empty (number of vertices is zero), then stop. No camera rotation is possible that would make all displacement vectors intersect the given FOE-region. Repeat the process using a larger FOE-region.

Figures 10(a-c) show the changing shape of the rotation polygon during the application of this process to the three displacement vectors in Figure 8.

Since the mapping from rotation space to the image plane is nonlinear (equation 9), the straight lines between vertices in the rotation polygon do not correspond to straight lines in the image. They are, however, approximated as straight lines in order to simplify the intersection with the open triangle. The dotted lines in the image plane show the *actual* mapping of the rotation polygon onto the image. It can be seen that the deviations from straight lines are small and can be neglected.

Figure 10(c) shows the final rotation polygon after examining the three displacement vectors. The amount of actual camera rotation (θ=–2.0°,ϕ=5.0°) is marked with a small circle (arrow).

Of course, increasing the number of displacement vectors improves the rotation estimate. In practice, the amount of camera rotation can be constrained to a range of below 1° in both directions. It is interesting, although not surprising, that rotation can be estimated more accurately when the displacement vectors are short, i.e., when the amount of camera translation is small. This is in contrast to estimating camera translation which is easier with long displacement vectors.

The situation when the rotation polygon becomes *empty* requires some additional considerations. As mentioned earlier, in such a case no camera rotation is possible that would make all displacement vectors pass through the given FOE-region. This could indicate one of the two alternatives:

- At least one of the displacement vectors belongs to a moving object.
- The given FOE-region does not contain the actual location of the FOE, i.e., the region is *not feasible*.

The latter case is of particular importance. If a region can be determined *not* to contain the FOE, then the FOE must necessarily lie outside this region. Therefore, the above method can not only be used to estimate the amount of camera rotation, but also to search for the location of the FOE. Unfortunately, if the rotation polygon does not become empty, this does *not* imply that the FOE is actually inside the given region. It only means that all displacement vectors would *pass through* this region, not that they have a common *intersection* inside this region. However, if not all vectors pass through a certain region, then this region cannot possibly contain the FOE. The following recursive algorithm searches a given region for the FOE by splitting it into smaller pieces (divide-and-conquer):

```
MIN-FEASIBLE (region, min-size, disp-vectors):
  if SIZE (region) < min-size then return (region)
  else
    if FEASIBLE (region, disp-vectors) then
      return (union (
        MIN-FEASIBLE (sub-region-1, min-size,
            disp-vectors),
        MIN-FEASIBLE (sub-region-2, min-size,
            disp-vectors),
        ....
        MIN-FEASIBLE (sub-region-n, min-size,
            disp-vectors)))
    else return (nil) {region does not contain the FOE}
```

This algorithm searches for the smallest feasible FOE-region by systematically discarding subregions from further consideration. For the case that the shape of the original region is a square, subregions can be obtained by splitting the region into four subsquares of equal size.

The simple version shown here performs a depth-first search down to the smallest subregion (limited by the parameter "min-size"), which is neither the most elegant nor the most efficient approach. The algorithm can be significantly improved by applying a more sophisticated strategy, for example, by trying to discard subregions around the perimeter first before examining the interior of a region.

Two major problems were encountered with this method. *First*, the algorithm is computationally expensive since the process of computing feasible rotations must be repeated for every subregion. *Second*, a small region is more likely to be discarded than a larger one. However, when the size of the region becomes too small, errors induced by noise, distortion, or point-tracking may prohibit displacement vectors from passing though a region which actually contains the FOE.

Although this algorithm is not employed in the further treatment, it suggests an interesting alternative which departs significantly from traditional FOE-algorithms. Its main attractiveness is that it is inherently region-oriented in contrast to most other techniques which search for a single FOE-location. For the purpose of estimating the amount of rotation, the method using points at far distance mentioned earlier is probably more practical. Two other alternatives for locating the FOE once the rotation components have been estimated are discussed in the following.

3.2.2 Locating the FOE in a partially derotated image

After applying a particular derotation mapping to the displacement field, the question is how close the new displacement field to a *radial mapping*, where all vectors diverge from one image location. If the displacement field is really radial, then the image is completely derotated and only the components due to camera translation remain. Two different methods for measuring this property are discussed. One method uses the *Variance of Intersection* at imaginary horizontal and vertical lines. The second method computes the *Linear Correlation Coefficient* to measure how "radial" the displacement field is.

A. Variance of Intersection. Prazdny[12] suggests to estimate the disturbance of the displacement field by computing the variance of intersections of one displacement vector with all other vectors. If the intersections lie in a small neighborhood, then the variance is small, which indicates that the displacement field is almost radial.

The problem can be simplified by using an imaginary horizontal and vertical line instead, whose orientation is not affected by different camera rotations. Figure 11 shows 5 displacement vectors $\mathbf{P}_1 \rightarrow \mathbf{P}_1' \ldots \mathbf{P}_5 \rightarrow \mathbf{P}_5'$ intersecting a vertical line at x at $\bar{y}_1 \ldots \bar{y}_5$. Moving the vertical line from x towards x_0 will bring the points of intersection closer together and will thus result in a smaller variance. The point of intersection of a displacement vector $\mathbf{P}_i \rightarrow \mathbf{P}_i'$ with a vertical line at x is given by

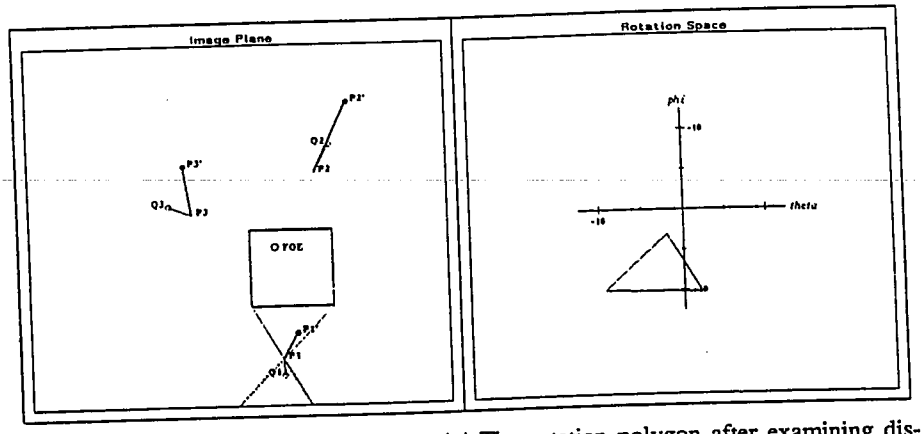


Figure 10: Changing rotation polygon. *(a)* The rotation polygon after examining displacement vector **P1→P1′**. Any camera rotation *inside* the polygon would move the endpoint of the displacement vector (P1′) into the open triangle formed by the tangents through **P1** to the maximal FOE-region given by the square in the image plane. The actual mapping of the rotation polygon into the image plane is shown with a dotted outline.

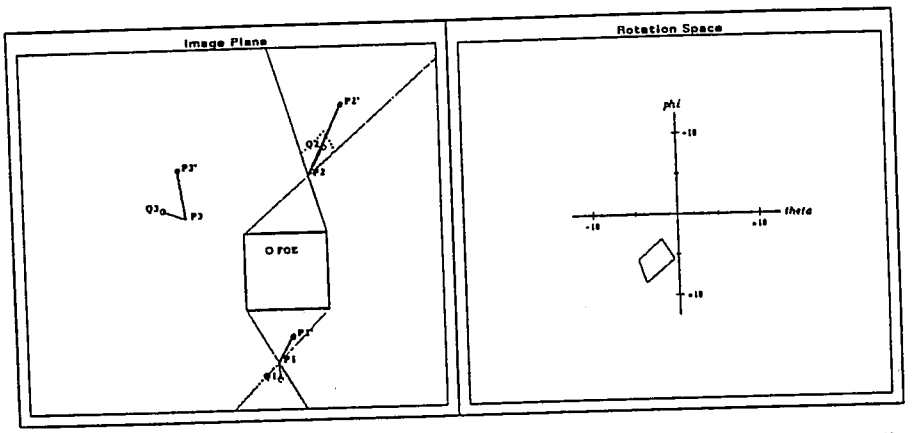


Figure 10: Changing rotation polygon. *(b)* The rotation polygon after examining displacement vectors **P1→P1′** and **P2→P2′**.

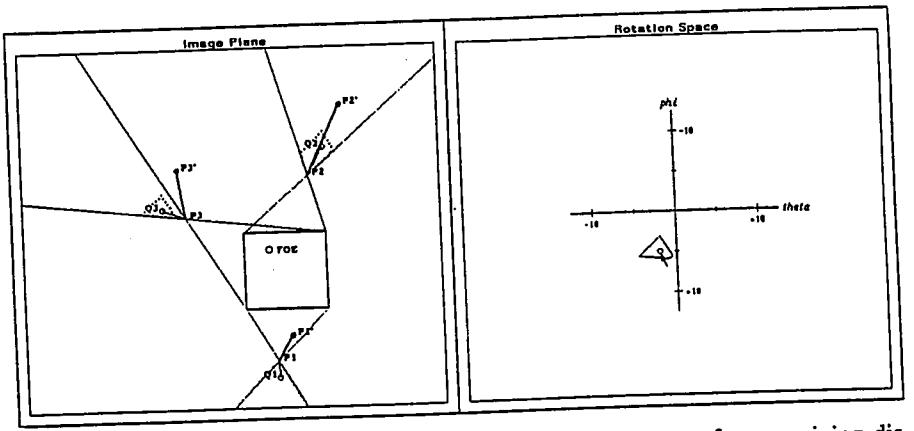


Figure 10: Changing rotation polygon. *(c)* The rotation polygon after examining displacement vector (**P1→P1′**, **P2→P2′**, and **P3→P3′**). **P1→P1'**, **P2→P2'** and **P3→P3'**. The amount of actual camera rotation is marked with a small circle (arrow).

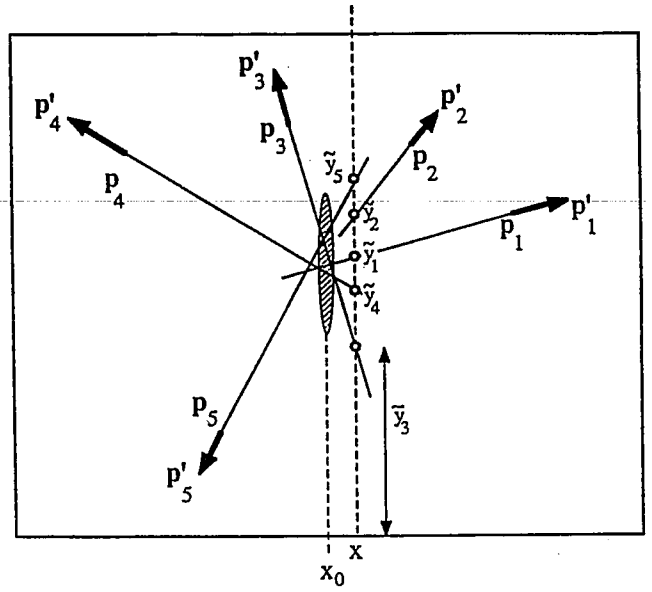


Figure 11: Intersecting the displacement vectors with a vertical line at x. When the vertical line is moved towards x_0 the points of intersection move closer together and therefore the *variance* of intersection decreases.

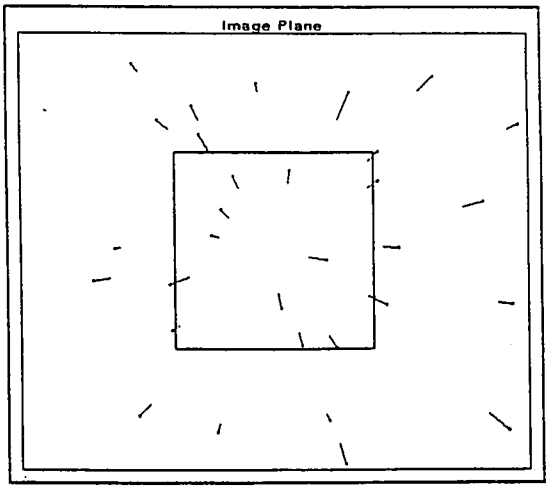


Figure 12: Displacement field used to evaluate various error functions. The square (±100 pixels in both directions) marks the region of evaluation.

$$\tilde{y}_i = \frac{x_i y_i' - y_i x_i'}{x_i - x_i'} \quad . \tag{28}$$

The variance of intersection of all displacement vectors with the vertical line at position x is

$$\sigma^2(x) = \frac{1}{N}\left[\sum_{i,\, x_i \neq x_i'} \tilde{y}_i^2 - \frac{1}{N}\left[\sum_{i,\, x_i \neq x_i'} \tilde{y}_i\right]^2\right] . \tag{29}$$

To find the vertical cross section with minimum intersection variance, the first derivative of (29) with respect to x is set to zero. The location x_0 of minimum intersection variance is then obtained.[1] Similarly, the position of a horizontal cross section with minimal intersection variance can be obtained.

The square root of the variance of intersection (standard deviation) at a vertical line was evaluated on the synthetic displacement field shown in Figure 12. The actual FOE is located in the center of the image. The square around the center (±100 pixels in both directions) marks the region over which the error functions are evaluated.

Figure 13 shows the distribution of the intersection standard deviation for increasing residual rotations in vertical direction in the absence of noise. Locations of displacement vectors are represented by real numbers (not rounded to integer values).

In Figure 13(a), no residual rotation exists, i.e., the displacement field is perfectly radial. The value of the horizontal position of the cross section varies ±100 pixels around the actual FOE. The standard deviation is zero for $x = x_f$ (the x-coordinate of the FOE) and increases linearly on both sides of the FOE. In Figures 13(b-d), the residual vertical rotation is increased from 0.2° to 1.0°. The bold vertical bar marks the horizontal position of minimum standard deviation, the thin bar marks the location of the FOE. It can be seen that the *amount* of minimum standard deviation rises with increasing disturbance by rotation, but that the *location* of minimum standard deviation does not necessarily move away from the FOE.

Figures 14-16 show the same function under the influence of noise. In Figure 14, noise was applied by merely rounding the locations of displacement vectors to their nearest integer values. Uniform noise of ±1 and ±2 pixels was added to image locations in Figures 15 and 16. It can be seen that the effects of noise are similar to the effects caused by residual rotation components. The purpose of this error function is to determine (a) where the FOE is located, and (b) how "radial" the current displacement field is.

If the displacement field is already perfectly derotated, then the location of minimum intersection standard deviation is, of course, the location of the FOE. Ideally all vectors pass through the FOE, such that a cross section through the FOE yields zero standard deviation. The question is how well the FOE can be located in an image which is *not* perfectly derotated.

Figure 17 plots the *location* of minimum intersection standard deviation under varying horizontal rotation. The vertical rotation is kept fixed for each plot. Horizontal camera rotations from -1° to +1° are shown on the abscissa (*rot*). The ordinate (*x0*) gives the location of minimum standard deviation in the range of ±100 pixels around the FOE (marked *xf*). It is not surprising that the location of minimum standard deviation depends strongly on the amount of horizontal rotation.

The problem is, however, that the location of minimum standard deviation is not necessarily closer to the FOE when

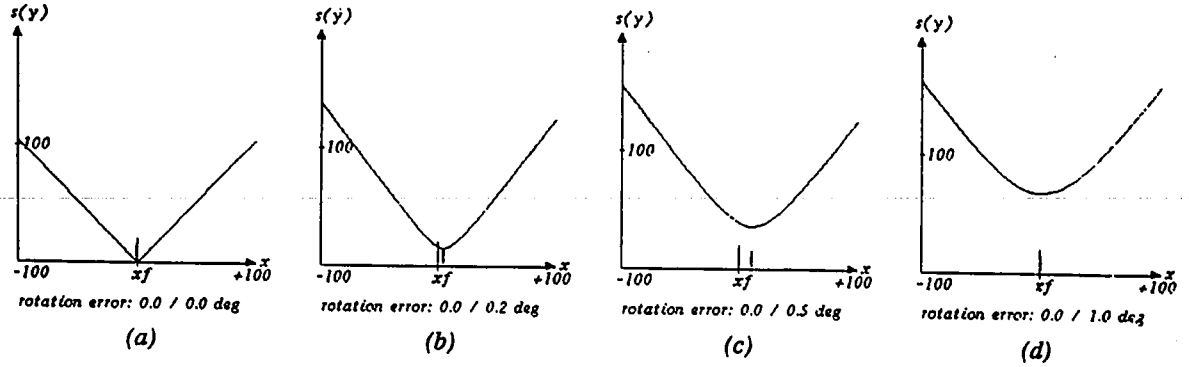


Figure 13: Standard deviation of intersection at a vertical cross section at position x for different amounts of vertical rotation. *(a)* Without vertical rotation, *(b)* with 0.2° vertical rotation, *(c)* 0.5° and *(d)* 1.0°. The horizontal rotation is 0° in all cases. No noise was applied and image positions were not rounded to integers. The error values are shown for ±100 pixels around the x-coordinate of the FOE (x_f), which is marked with a thin bar. The location of minimum standard deviation is marked with a thick bar.

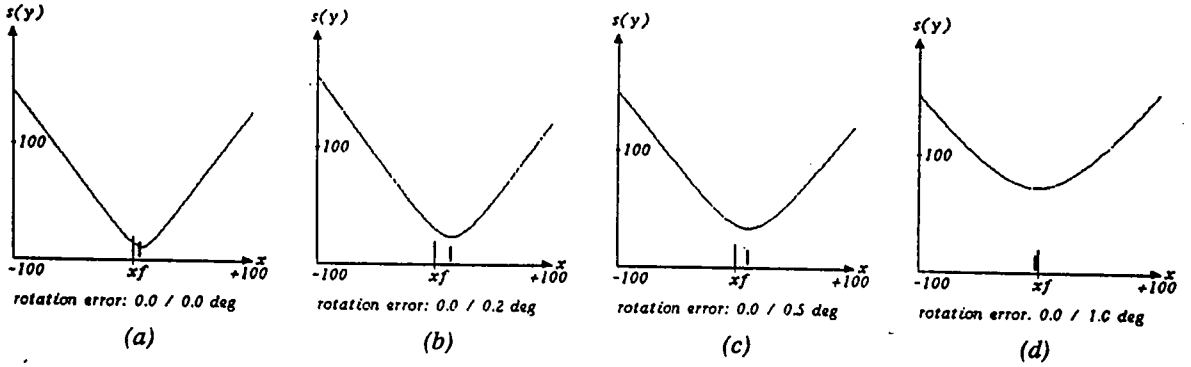


Figure 14: Standard deviation of intersection (square root) at a vertical cross section at position x for different amounts of vertical rotation with no horizontal rotation. *(a)* Without vertical rotation, *(b)* with 0.2° vertical rotation, *(c)* 0.5°, and *(d)* 1.0°. No noise was applied and image positions were rounded to the closest integer values.

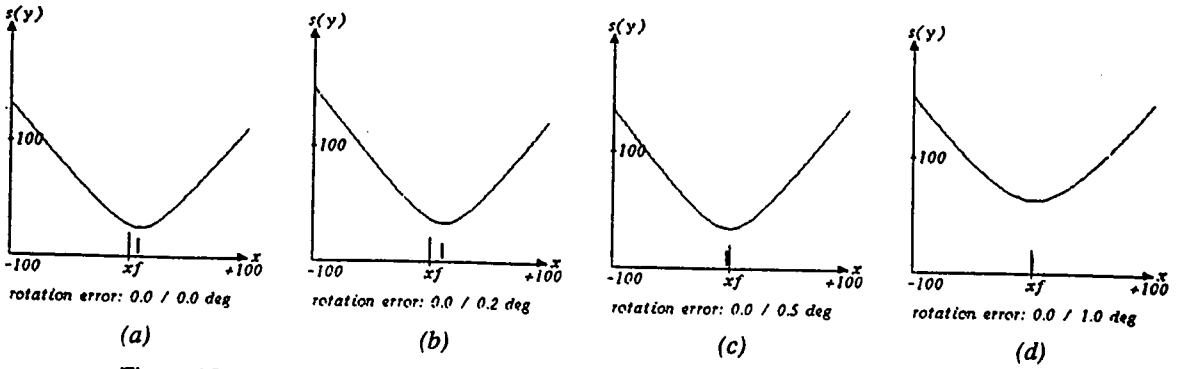


Figure 15: Standard deviation of intersection (square root) at a vertical cross section at position x for different amounts of vertical rotation with no horizontal rotation. *(a)* Without vertical rotation, *(b)* with 0.2° vertical rotation, *(c)* 0.5°, and *(d)* 1.0°. Uniform noise of ±1 pixels was applied to the image locations.

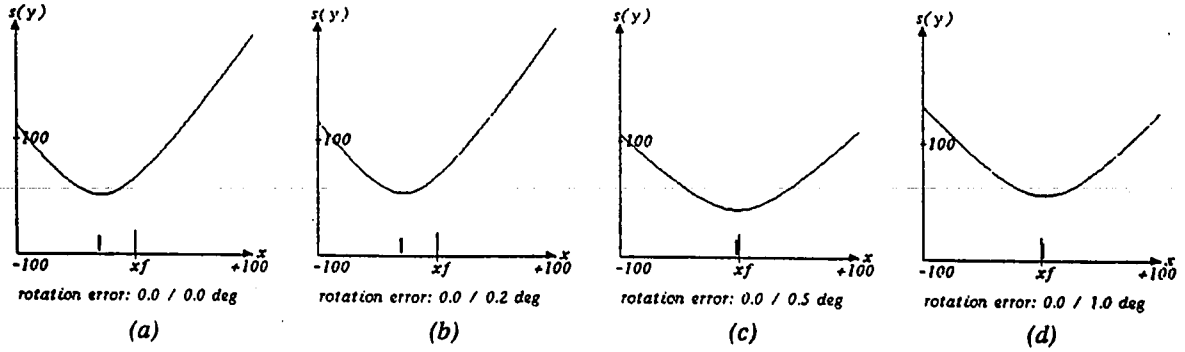


Figure 16: Standard deviation of intersection (square root) at a vertical cross section at position x for different amounts of vertical rotation with no horizontal rotation. *(a)* Without vertical rotation, *(b)* with 0.2° vertical rotation, *(c)* 0.5°, and *(d)* 1.0°. Uniform noise of ±2 pixels was applied to the image locations.

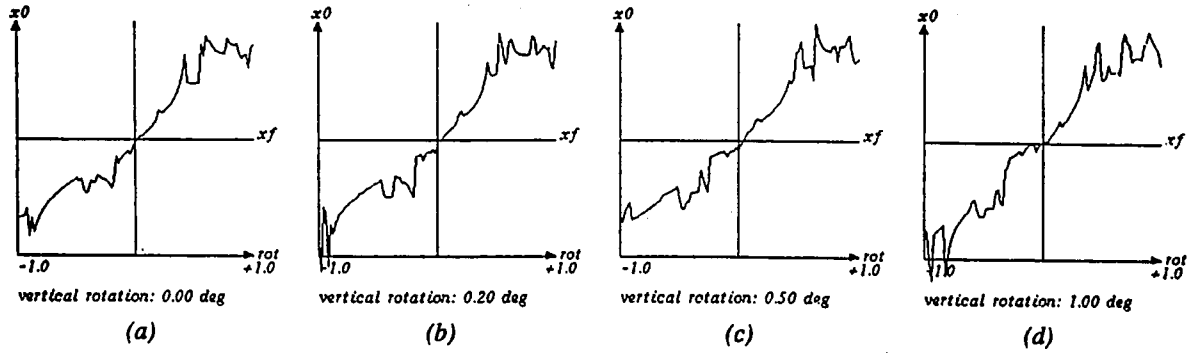


Figure 17: *Location* of minimum intersection standard deviation under varying horizontal rotation. The amount of vertical rotation is kept fixed in each plot. *(a)* Without vertical rotation, *(b)* with 0.2° vertical rotation, *(c)* 0.5°, and *(d)* 1.0°. Image locations were digitized but no noise was added. The horizontal location of the FOE is marked *xf*.

the amount of rotation is less. The function is only well behaved in a narrow range around zero rotation, which means that the estimate of the camera rotation must be very accurate to successfully locate the FOE.

The second purpose of this error function is to measure how "radial" the displacement field is after partial derotation. This should be possible by computing the *amount* of minimum intersection standard deviation. Intuitively, a smaller amount of minimum intersection standard deviation should indicate that the displacement field is less disturbed by rotation. Figure 18 and 19 show that this is generally true.

For the noise-free case in Figure 18, the amount of minimum intersection standard deviation becomes zero in the absence of horizontal and vertical rotations (a), indicating that the derotation is perfect. Unfortunately, the function is not well behaved even in this relatively small range of rotations (±1.0°). The curve exhibits some sharp local minima where an algorithm searching for an optimal derotation would get trapped easily. Figure 19 shows the same function in the presence of noise.

B. Linear Correlation. The second method of measuring how close a displacement field is to a radial pattern again uses the points of intersection at vertical (or horizontal) lines. The basic idea is illustrated in Figure 20. The displacement vectors are intersected by *two* vertical lines, both of which lie on the same side of the FOE. Since the location of the FOE is not known, the two lines are simply located at a sufficient distance from any possible FOE-location. This results in two sets of intersection points $\{(x_1,y_{1i})\}$ and $\{(x_2,y_{2i})\}$.

If all displacement vectors emanate from one single image location, then the distances between corresponding intersection points in the two sets must be proportional, i.e.,

$$\frac{y_{1i} - y_{1j}}{y_{2i} - y_{2j}} = \frac{y_{1j} - y_{1k}}{y_{2j} - y_{2k}} \quad \text{for all } i,j,k\,. \tag{30}$$

Therefore, a linear relationship exists between the vertical coordinates of intersection points on these two lines. The "goodness" of this linear relationship is easily measured by computing the *correlation coefficient* for the y-coordinates of the two sets of points.[1]

The resulting coefficient is a real number in the range from -1.0 to +1.0. If both vertical lines are on the same side of the FOE, then the optimal value is +1.0. Otherwise, if the FOE lies between the two lines, the optimal coefficient is -1.0. The horizontal position of the two vertical lines is of no importance, as long as one of these conditions is satisfied. For example, the left and right border lines of the image can be used.

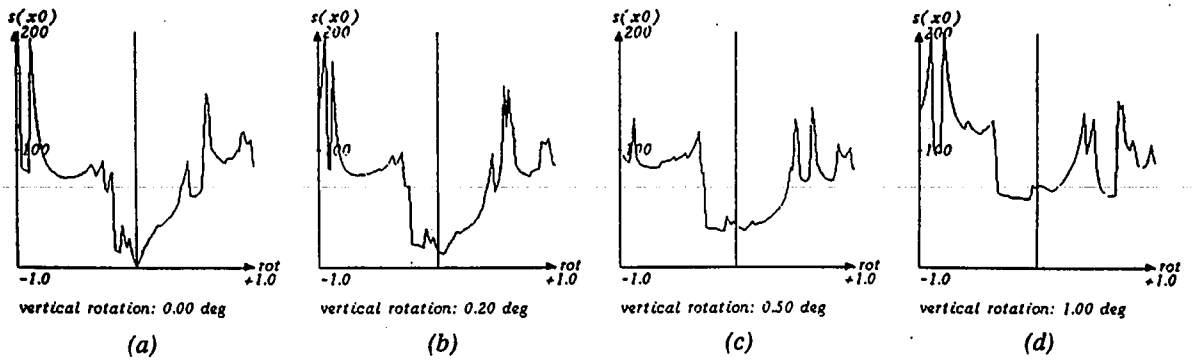


Figure 18: *Amount* of minimum intersection standard deviation under varying horizontal rotation. The amount of vertical rotation is kept fixed in each plot. *(a)* Without vertical rotation, *(b)* with 0.2° vertical rotation, *(c)* 0.5°, and *(d)* 1.0°. Image locations were digitized but no noise was added.

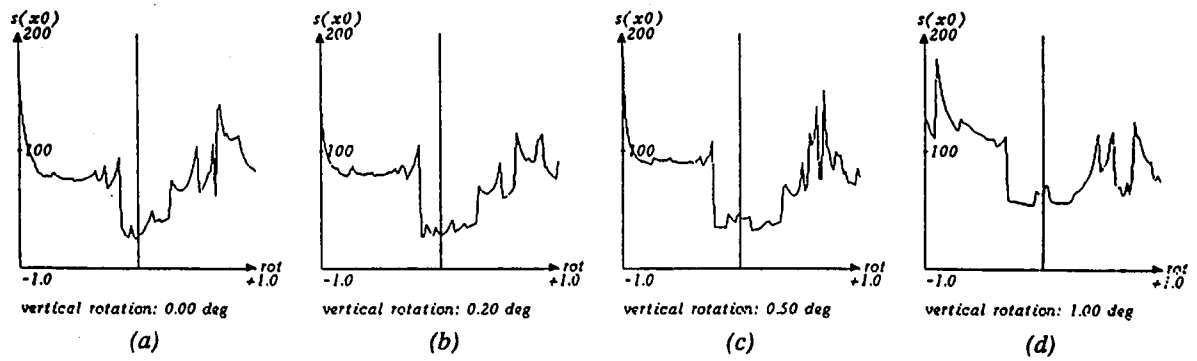


Figure 19: *Amount* of minimum intersection standard deviation under varying horizontal rotation as in Figure 18. *(a)* Without vertical rotation, *(b)* with 0.2° vertical rotation, *(c)* 0.5°, and *(d)* 1.0°. Uniform noise of ±2 pixels was added to image locations.

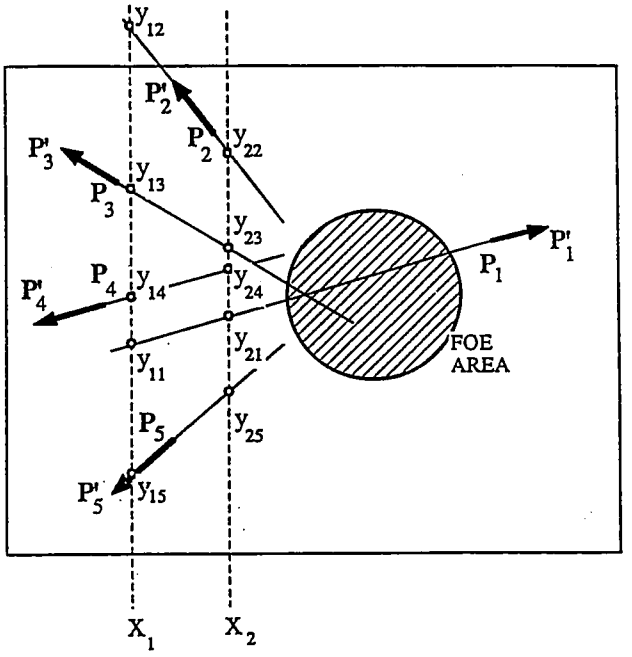


Figure 20: Intersecting displacement vectors with two vertical lines, both of which lie on the same side of the FOE.

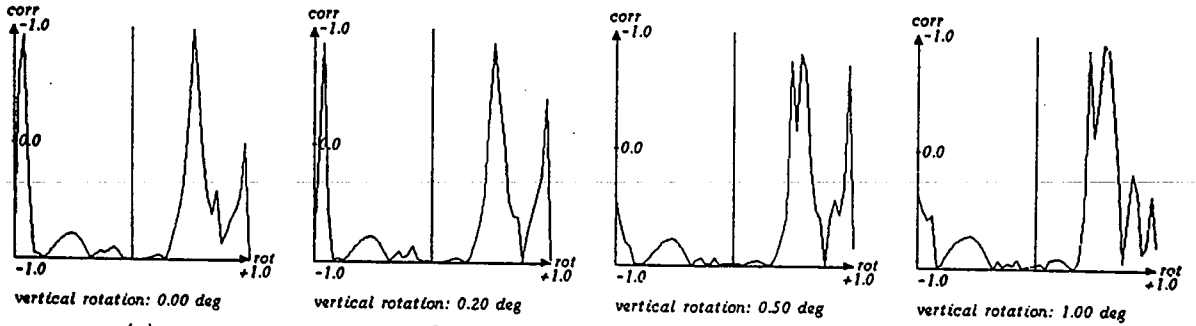


Figure 21 Correlation coefficient for the intersection of displacement vectors at two vertical lines under varying horizontal rotations in the noise-free case. *(a)* Without vertical rotation, *(b)* with 0.2° vertical rotation, *(c)* 0.5°, and *(d)* 1.0°. The optimal coefficient is +1.0 (horizontal axis).

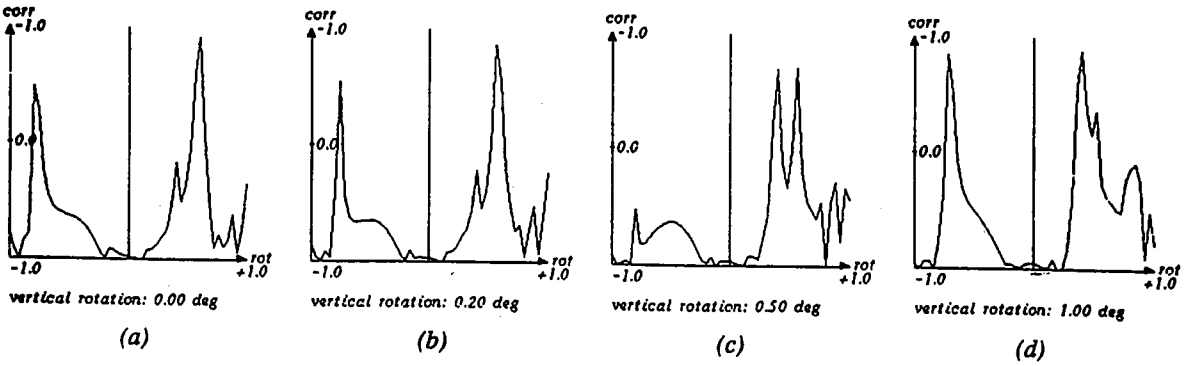


Figure 22 Correlation coefficient for the intersection of displacement vectors at two vertical lines under varying horizontal rotations. Uniform noise of ±2 pixels was added to image locations. Without vertical rotation *(a)*, with 0.2° vertical rotation *(b)*, 0.5° *(c)* and 1.0° *(d)*. The optimal coefficient is +1.0 (horizontal axis).

Figures 21 and 22 show plots for this correlation coefficient under the same conditions as in Figures 18 and 19. No noise was applied for Figure 21. The shapes of the curves are similar to those for the minimum standard deviations shown earlier, with peaks at the same locations. It is apparent, however, that each curve has several locations where the coefficient is close to the optimum value (+1.0), i.e., no distinct global optimum exists which is not only the case in the presence of noise (Figure 22). This fact makes the method of maximizing the correlation coefficient useless for computing the FOE.

3.3 Rotation from FOE

The main problem encountered in computing the FOE in section 3.2 was that none of the functions examined was well behaved, making the search for an optimal derotation and the location of the FOE difficult. Disturbances induced by noise and residual rotation components are amplified by extending short displacement to straight lines and computing their intersections. The method described below avoids this problem by guessing an FOE-location first and estimating the optimal derotation for this particular FOE in the second step.

Given the two images $\mathbf{I}_0$ and $\mathbf{I}_1$ of corresponding points, the main algorithmic steps of this approach are:

(1) Guess an FOE-location $\mathbf{x}_f^{(i)}$ in image $\mathbf{I}_0$ (for the current iteration i).

(2) Determine the derotation mapping $\mathbf{r}_\theta^{-1}$, $\mathbf{r}_\phi^{-1}$ which would transform image $\mathbf{I}_1$ into an image $\mathbf{I}_1'$ such that the mapping $(\mathbf{x}_f^{(i)}, \mathbf{I}_0, \mathbf{I}_1')$ deviates from a radial mapping (23) with minimum error $E^{(i)}$.

(3) Repeat steps *(1)* and *(2)* until an FOE-location $\mathbf{x}_f^{(k)}$ with the lowest minimum error $E^{(k)}$ is found.

An initial *guess* for the FOE-location is obtained from knowledge about the orientation of the camera with respect to the vehicle. For subsequent pairs of frames, the FOE-location computed from the previous pair can be used as a starting point.

Once a particular $\mathbf{x}_f$ has been selected, the problem is to compute the rotation mappings $\mathbf{r}_\theta^{-1}$ and $\mathbf{r}_\phi^{-1}$ which, when applied to the image $\mathbf{I}_1$, will result in an optimal radial mapping with respect to $\mathbf{I}_0$ and $\mathbf{x}_f$.

To measure how close a given mapping is to a radial mapping, the perpendicular distances between points in the second image ($\mathbf{x}_i'$) and the "ideal" displacement vectors is measured. The "ideal" displacement vectors lie on straight lines passing through the the FOE $\mathbf{x}_f$ and the points in the first image $\mathbf{x}_i$ (Figure 23). The sum of the squared perpendicular distances d_i is the final error measure. For each set of corresponding image points ($\mathbf{x}_i \in \mathbf{I}$, $\mathbf{x}_i' \in \mathbf{I}'$), the error measure is defined as

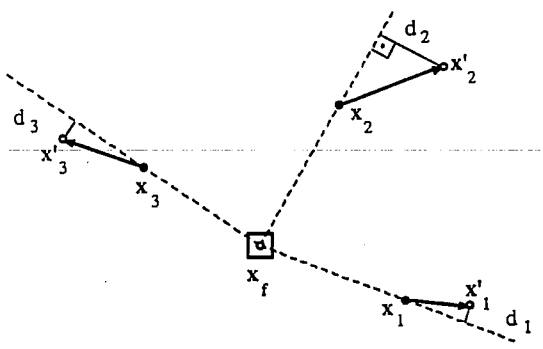


Figure 23: Measuring the perpendicular distance d_i between lines from $\mathbf{x}_f$ through points $\mathbf{x}_i$ and points $\mathbf{x}_i'$ in the second image.

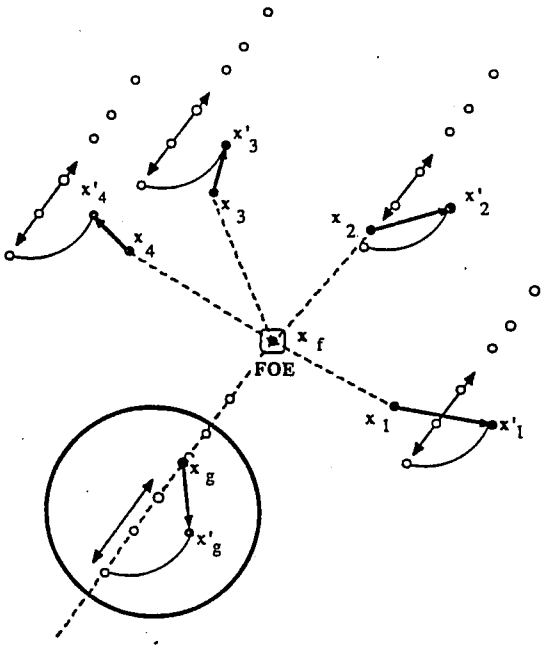


Figure 24: One vector $\mathbf{x}_g$ is selected from the set of displacement vectors to determine the optimum 2-D shift to be applied to points $\mathbf{x}_i'$, given a FOE-location $\mathbf{x}_f$. First $\mathbf{x}_g'$ is forced onto the line $\overline{\mathbf{x}_f\mathbf{x}_g}$ and then the entire image $\mathbf{I}' = \{\mathbf{x}_1', \mathbf{x}_2', ...\}$ is translated in the direction of this line until the error value reaches a minimum.

$$E(\mathbf{x}_f) = \sum_i E_i = \sum_i d_i^2 = \sum_i \left[\frac{1}{|\vec{\mathbf{x}_f\mathbf{x}_i}|} \vec{\mathbf{x}_f\mathbf{x}_i} \times \vec{\mathbf{x}_f\mathbf{x}_i'} \right]^2 . \quad (31)$$

In the following, it is assumed that the amount of residual image rotation in horizontal and vertical direction is moderately small (less than 4°). 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 (as described earlier) and applied to the image before the FOE computation. With small amounts of rotation, the actual rotation mapping, where points move on horizontal and vertical hyperbolic paths, can be approximated by a horizontal and vertical *shift* with constant length over the entire image.

Under this condition, the inverse rotation mapping r_ϕ^{-1}, r_θ^{-1} can be approximated by a adding a constant vector $\mathbf{s} = (s_x\ s_y)$ which is independent of the image location:

$$\mathbf{I}_1' = r_\theta^{-1} r_\phi^{-1} \mathbf{I}_1 \approx \mathbf{s} + \mathbf{I}_1 . \quad (32)$$

Given two images $\mathbf{I}$ and $\mathbf{I}'$ the error measure (31) becomes

$$E(\mathbf{x}_f, \mathbf{s}) = \sum_i \left\{ \frac{1}{|\vec{\mathbf{x}_f\mathbf{x}_i'}|^2} \left[\vec{\mathbf{x}_f\mathbf{x}_i} \times (\vec{\mathbf{x}_f\mathbf{x}_i'} + \mathbf{s}) \right]^2 \right\} \quad (33)$$

where $\mathbf{x}_i \in \mathbf{I}$ and $\mathbf{x}_i' \in \mathbf{I}'$. For a given FOE-location $\mathbf{x}_f$, the problem is to minimize E with respect to the two unknowns s_x and s_y. To reduce this problem to a one-dimensional search, one point $\mathbf{x}_g$, called the *Guiding Point*, is selected in image $\mathbf{I}$ which is forced to maintain zero error (Figure 24). Therefore, the corresponding point $\mathbf{x}_g'$ must lie on a straight line passing through $\mathbf{x}_f$ and $\mathbf{x}_g$ Any shift $\mathbf{s}$ applied to the image $\mathbf{I}'$ must keep $\mathbf{x}_g'$ on this straight line, so

$$\mathbf{x}_g' + \mathbf{s} = \mathbf{x}_f + \lambda (\mathbf{x}_g - \mathbf{x}_f) \quad \text{for all s,} \quad (34a)$$

and thus,

$$\mathbf{s} = \mathbf{x}_f - \mathbf{x}_g' + \lambda (\mathbf{x}_g - \mathbf{x}_f) \quad (\lambda \in R). \quad (34b)$$

For $\lambda = 1$, $\mathbf{s} = \mathbf{x}_g - \mathbf{x}_g'$ which is the vector $\mathbf{x}_g' \rightarrow \mathbf{x}_g$. This means that the image $\mathbf{I}'$ is shifted such that $\mathbf{x}_g$ and $\mathbf{x}_g'$ overlap. This leaves λ as the only free variable and the error function (33) is obtained as

$$E(\lambda) = \sum_i \left[\lambda A_i + B_i - C_i \right]^2 \quad (35)$$

with

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

$$A_i = \frac{1}{l_{if}} (y_i - y_f)(x_g - x_f) - (x_i - x_f)(y_g - y_f)$$

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

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

Differentiating 35 with respect to λ and forcing the resulting equation to zero yields the parameter for the optimal shift $\mathbf{s}_{opt}$ as

$$\lambda_{opt} = \frac{\sum A_i C_i - \sum A_i B_i}{\sum A_i^2} . \quad (36)$$

The optimal shift $\mathbf{s}_{opt}$ and the resulting minimum error $E(\lambda_{opt})$ for the given FOE-location $\mathbf{x}_f$ is obtained by inserting λ_{opt} into equations (34b) and (35) respectively, giving

$$E_{min}(\mathbf{x}_f) = \lambda_{opt}^2 \sum A_i^2 + 2\lambda_{opt} \left[\sum A_i B_i - \sum A_i C_i \right] - 2 \sum B_i C_i + \sum B_i^2 + \sum C_i^2 . \quad (37)$$

The normalized error E_n shown in the following results (Figures 26-31) is defined as

$$E_n(\mathbf{x}_f) = \sqrt{\frac{1}{N} E_{min}(\mathbf{x}_f)} \quad (38)$$

where N is the number of displacement vectors used for com-

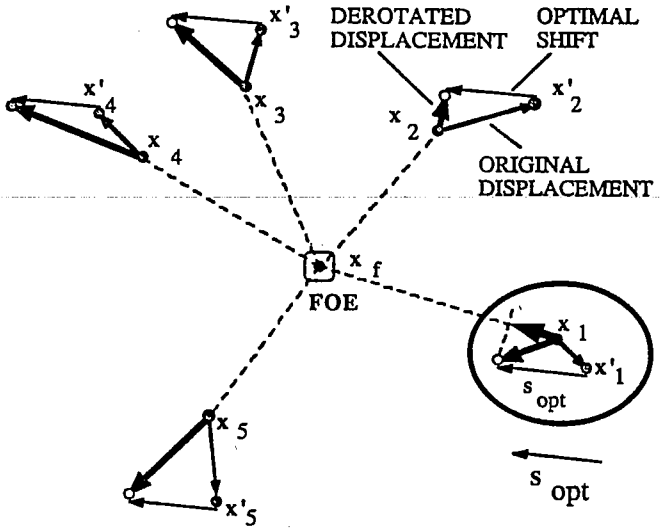


Figure 25: FOE-locations are *prohibited* if the displacement field resulting from the application of the optimal shift s_{opt} contains vectors pointing towards the FOE. This is the case at point $\mathbf{x}_1$.

puting 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 (Figure 24). If after applying $s_{opt}(\mathbf{x}_f)$ to the second image $\mathbf{I}'$, the resulting displacement field contains vectors pointing *towards* the hypothesized $\mathbf{x}_f$, then this FOE-location is *prohibited* and can be discarded from further consideration. Figure 25 shows a field of 5 displacement vectors. The optimal shift s_{opt} for the given $\mathbf{x}_f$ is shown as a vector in the lower right-hand corner. When s_{opt} is applied to point $\mathbf{x}_1'$, the resulting displacement vector (shown fat) does not point away from the FOE. Since its projection onto the line $\mathbf{x}_f\mathbf{x}_1$ points *towards* the FOE, it is certainly not consistent with a radial expansion pattern.

The final algorithm for determining the direction of heading as well as horizontal and vertical camera rotations is the following:

Find-FOE:

(1) Guess an initial FOE $\mathbf{x}_f^0$, for example the FOE-location obtained from the previous pair of frames.

(2) Starting from $\mathbf{x}_f^0$, search for a location $\mathbf{x}_f^{opt}$ where $E_{min}(\mathbf{x}_f^{opt})$ is a minimum. A technique of *steepest descent* is used, where the search proceeds in the direction of least error.

(3) Determine a region around $\mathbf{x}_f^{opt}$ in which the error is below some threshold.

The search for this FOE-*area* is conducted at FOE-locations lying on a grid of fixed width. In the examples shown, the grid spacing is 10 pixels on both *x*- and *y*-directions.

The error function $E(\mathbf{x}_f)$ 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)*.

3.4 Experiments on Synthetic Data

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.

Figure 26 shows the distribution of the normalized error $E_n(\mathbf{x}_f)$ for a sparse and relatively short displacement field containing 7 vectors. Residual rotation components of ±2° in horizontal and vertical direction are present in *(b)-(d)* to visualize their effects upon the image. This displacement field was used with different average vector lengths (indicated as *length-factor*) for the other experiments on synthetic data. 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 $\mathbf{x}_f\mathbf{x}_g$.

In Figure 26, 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. At each grid point, the amount of error is indicated by the size of the circle. Heavy circles indicate error values which are above a certain threshold. Those FOE-locations that would result in displacement vectors which point *towards* the FOE (as described earlier) are marked as prohibited (+). It can be seen that the shape of the 2D error function changes smoothly with different residual rotations over a wide area and exhibits its minimum close to the actual location of the FOE.

Figures 27 to 32 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 Figure 26.

Figure 27 shows how the shape of the error function depends upon the average length of the displacement vectors in the absence of any residual rotation or noise (except digitization noise). Clearly, the minimum of the error function becomes more distinct with increasing amounts of displacement.

Figure 28 shows the effect of increasing residual rotation in horizontal direction upon the shape of the error function.

Figure 29 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 Figure 29.

In Figure 30, residual rotations in both horizontal and vertical direction are present. It can be seen (Figure 30*(a-e)*) that the error function is quite robust against rotational components in the image. Figure 30*(f-j)* shows the amounts of optimal linear shift s_{opt} under the same conditions.

The result in Figure 30*(e)* shows the effect of large combined rotation of 4.0° / 4.0° in both directions. Here, the minimum of the error function is considerably off the actual

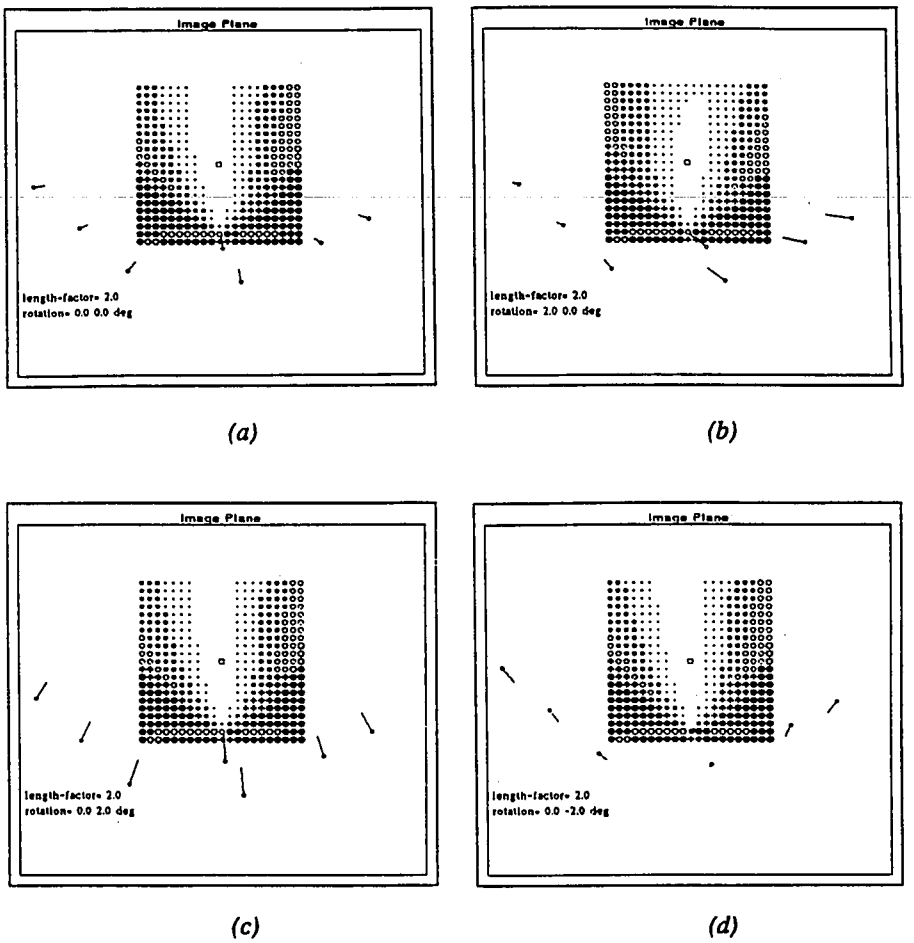


Figure 26: Displacement field and minimum error at selected FOE-locations. The shape of the error function is plotted over an area of ±100 pixels around the actual FOE, which is marked with a small square. The diameter of the circle drawn at each hypothesized FOE-location indicates the amount of normalized error (equation 40), large circles are locations of large error. 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).

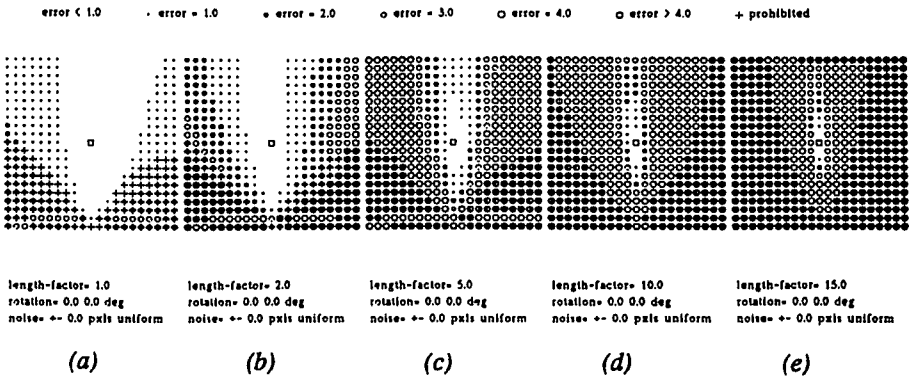


Figure 27 (a-e): Effects of increasing the average length of displacement vectors upon the shape of the error function. Length factors vary from 1 to 15. The error function was evaluated over the same image area of 200×200 pixels around the actual FOE (square) as in Figure 26. No rotation or noise was applied.

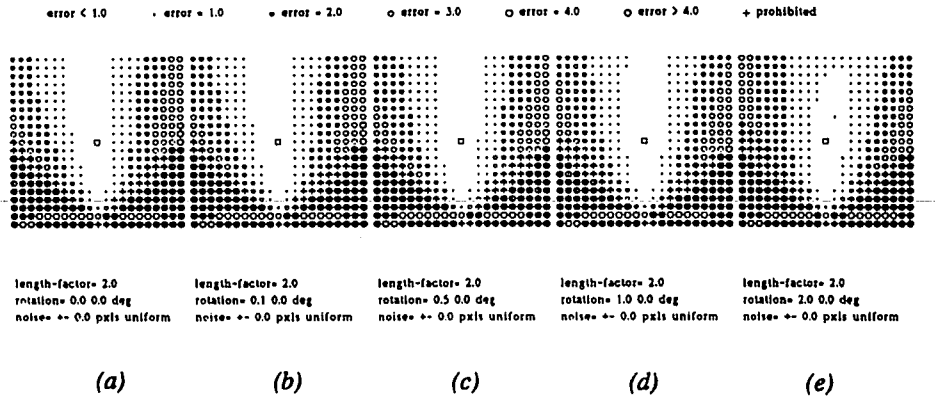


Figure 28 (a-e): Effects of increasing residual rotation in horizontal direction upon the shape of the error function for relatively short vectors (length-factor 2.0). No noise was applied.

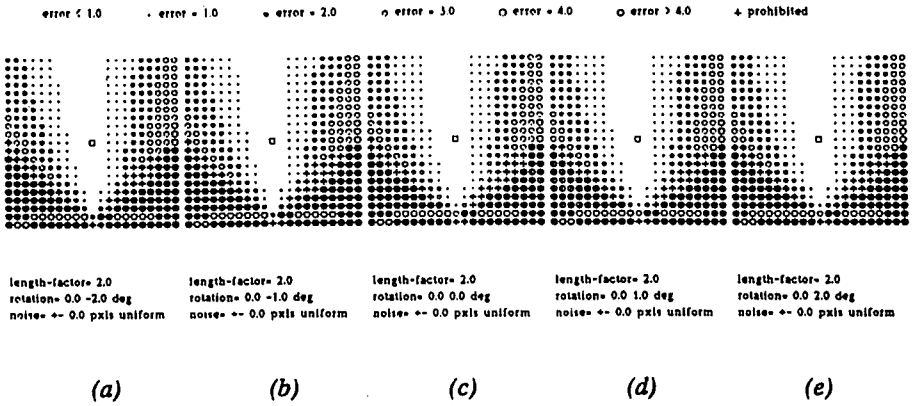


Figure 29 (a-e): Effects of increasing residual rotation in vertical direction upon the shape of the error function for relatively short vectors (length-factor 2.0). No noise was applied.

location of the FOE because of the error induced by using a linear shift to approximate the nonlinear 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 amounts of noise are shown in Figure 31. 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 7 vectors, the results do not provide information about the statistical effects of image noise. This would require more extensive modeling and simulation. However, what can be observed here is that the absolute minimum error increases with the amount of noise. It can thus serve as an indicator for the amount of noise present in the image and the reliability of the final result.

Again, 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. Figure 32 shows the error functions for two displacement fields with different average vector lengths. For the shorter displacement field (*length-factor* 2.0) in Figure 32*(a)*, the shape of the error function changes dramatically under the same amount of noise (compare Figure 30*(a)*). A search for the minimum error would inevitably converge towards a point indicated by the small arrow, far off the actual FOE. For the image with *length-factor* 5.0 (Figure 32*(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 the Figure 31*(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.

The performance of this FOE algorithm is shown in section 5.1 on a sequence of real images taken from the moving ALV. In the following section, 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. Given the absolute velocity of the vehicle, the absolute *distance* from the camera of 3D points in the scene can be estimated using equation (20).

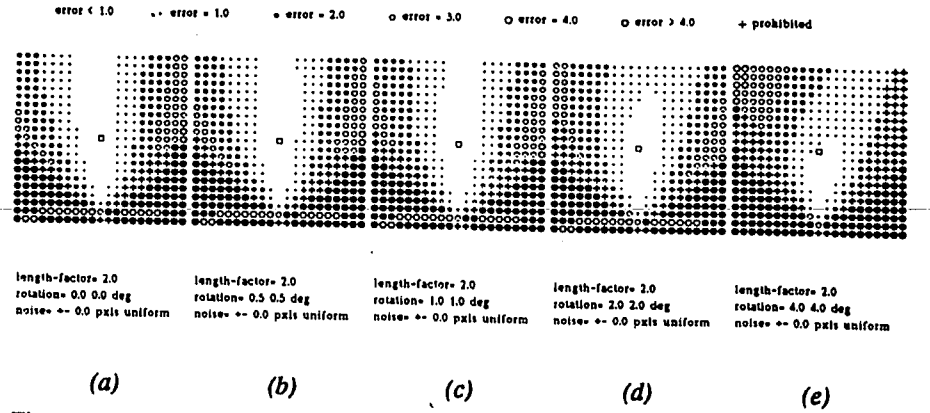


Figure 30 (a-e): Effects of increasing residual rotation in horizontal and vertical direction upon the shape of the error function for relatively short vectors (length-factor 2.0). No noise was applied.

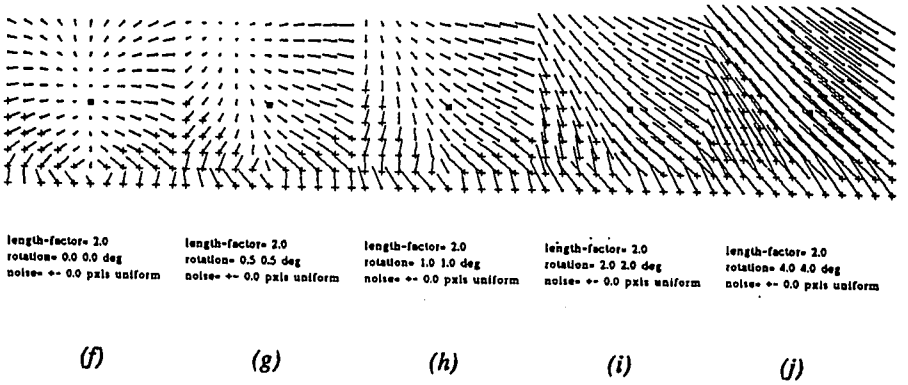


Figure 30 (f-j): The amount of optimal linear shift obtained under the same conditions as in Figure 30 *(a-e)*.

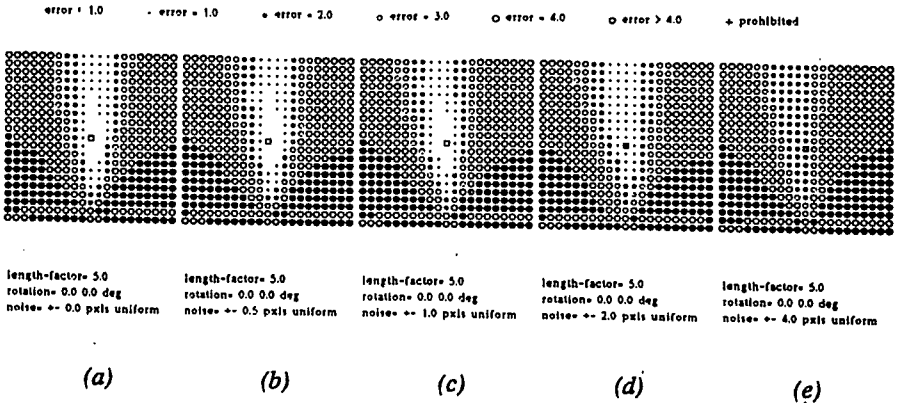
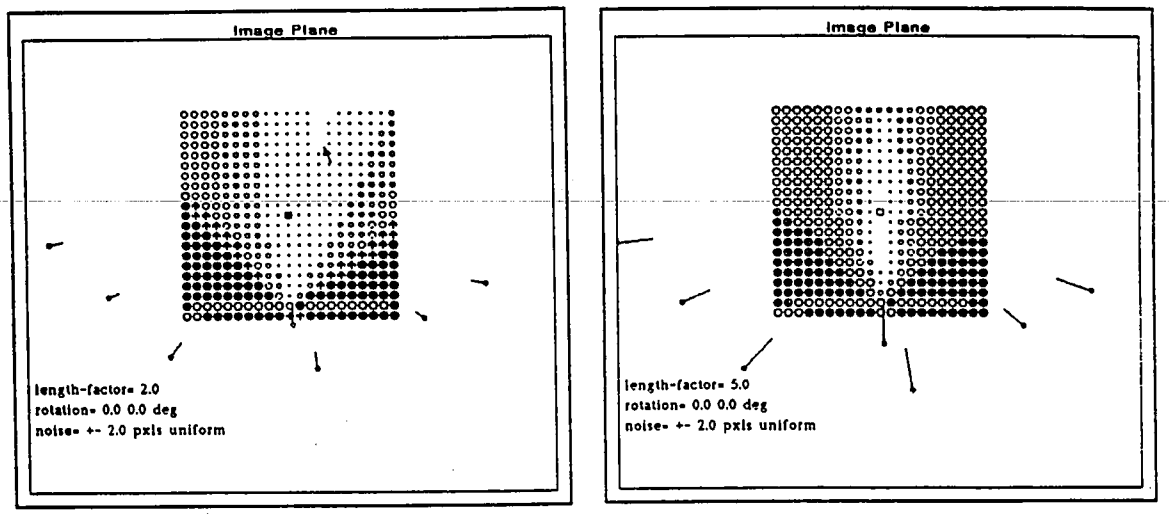


Figure 31 (a-e): The effects of uniform noise applied to image point coordinates for a constant average vector length. The shape of the error function become flat around the local minimum of the FOE with increasing levels of noise.



(a) *(b)*

Figure 32: 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.

3.4 Computing Velocity Over Ground

After the FOE has been computed following the steps outlined in the previous section, the direction of vehicle translation and the amount of rotation are known. From the derotated displacement field and the location of the FOE, the 3D layout of the scene can be obtained up to a common scale factor (20). As pointed out earlier, this scale factor and, consequently, the velocity of the vehicle can be determined if the 3D position of one point in space is known. Furthermore, it is easy to show[6,4] 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.

Since 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 Figure 33). 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 should be 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 (X Y Z) is 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 equations (14) and (15):

$$\theta_f = -\tan^{-1}\frac{x_f}{f} \tag{39}$$

$$\phi_f = -\tan^{-1}\left[y_f \frac{f^2}{(f^2 + x_f^2)\, f^2 - x_f^2\, y_f^2}\right]. \tag{40}$$

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

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \kappa \begin{bmatrix} \cos\theta_f & \sin\theta_f \sin\phi_f & -\sin\theta_f \cos\phi_f \\ 0 & \cos\phi_f & \sin\phi_f \\ \sin\theta_f & -\cos\theta_f \sin\phi_f & \cos\theta_f \cos\phi_f \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} \tag{41}$$

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

$$\kappa_s = \frac{-h}{y_s \cos\theta_f + f \sin\theta_f} \tag{42}$$

and its 3D distance is found by inserting $\mathbf{v}_s$ into equation 41 as

$$Z_s = -h\, \frac{x_s \sin\theta_f - y_s \cos\theta_f \sin\phi_f - f \cos\theta_f \cos\phi_f}{y_s \cos\phi_s + f \sin\phi_s}. \tag{43}$$

If a point on the ground is observed at two instances of time, $\mathbf{x}_s$ at time t and $\mathbf{x}_s'$ at t', the resulting distances from the vehicle Z_s at t and Z_s' at t' yield the amount of advancement

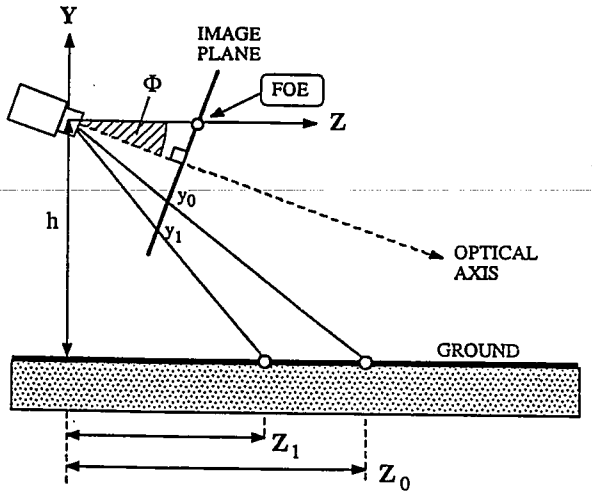


Figure 33: Side view of the camera traveling parallel to a flat surface. The camera advanced in direction Z, such that a 3D point on the ground moves relative to the camera from Z_0 to Z_1. The depression angle ϕ can be found from the location of the FOE in the image. The height of the camera above the ground is given.

$\Delta Z_s\ (t,t')$ and estimated velocity $V_s\ (t,t')$ in this period as

$$\Delta Z_s\ (t,t') = Z_s - Z_s' \tag{44}$$

$$V_s\ (t,t') = \frac{Z_s - Z_s'}{t' - t}\ . \tag{45}$$

Of course, image noise and tracking errors have a large impact upon the quality of the final velocity estimate. Therefore, the 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 $|\mathbf{x}_i - \mathbf{x}_i'|$ in the image is used as the weight for its contribution to the final result. Given a set of suitable displacement vectors $S = \{\ \mathbf{x}_i - \mathbf{x}_i'\ \}$, the estimate of the distance traveled by the vehicle is taken as the weighed average of the measurements ΔZ_i on individual vectors

$$\tilde{\Delta Z}\ (t,t') = \frac{\sum\ (|\mathbf{x}_i - \mathbf{x}_i'|\ \Delta Z_i)}{\sum |\mathbf{x}_i - \mathbf{x}_i'|} \tag{46}$$

and the final estimate for the vehicle velocity is

$$\tilde{V}\ (t,t') = \frac{\tilde{\Delta Z}}{t' - t}\ . \tag{47}$$

This computation was applied to a sequence of real images which is described in section 5.

4. THE QUALITATIVE SCENE MODEL

The choice of a suitable scheme for the internal representation of the scene is of great importance. The *Qualitative Scene Model* (QSM) is a 3D camera-centered interpretation of the scene that is built incrementally from visual information gathered over time. The nature of this model, however, is *qualitative* rather than a precise geometric description of the scene. The basic building blocks of the QSM are *entities*, which are the 3D counterparts of the 2D *features* observed in the image. For example, the point feature *A* located in the image at *x,y* at time *t*

(Point_Feature *A t x y*)

has its 3D counterpart in the model as

(Point_Entity *A*).

Since the model is camera-centered ("retinocentric"), the image locations and 2D movements of features are implicitly part (i.e., known facts) of the model. Additional entries are the properties of entities (e.g., "stationary" or "mobile") and relationships between entities (e.g. "closer"), which are not given facts but hypotheses about the real scene. This is expressed in the model as either

(Stationary *entity*) or (Mobile *entity*) .

It is one of the key features of the QSM that it generally contains not only one interpretation of the scene, but a (possibly empty) *set* of interpretations which are all pursued simultaneously. At any point in time, a hypothesis is said to be "feasible" if it exists in the QSM and is not in conflict with some observation made since it was established.

Interpretations are structured as an inheritance network of partial hypotheses. Individual scene interpretations are treated as "closed worlds", i.e., a new conclusion only holds within an interpretation where all the required premises are true. Interpretations are also checked for internal consistency, e.g., entities cannot be both stationary *and* mobile within the same interpretation.

The QSM is maintained through a generate-and-test process as the core of a rule-based blackboard system. The two major groups of rules are: *Generation Rules* and *Verification Rules*.

Generation Rules

Generation rules examine the (derotated) image sequence for significant changes and modify each interpretation in the QSM. Some of these observations have unconditional effects upon the model. For example, if an image feature is found to be moving *towards* the Fuzzy FOE (instead of diverging away from it), then it belongs to a moving entity in 3D space. The actual rule contains only one premise and asserts (MOBILE ?x) as a global fact (i.e., it is true in every interpretation):

```
(defrule DEFINITE_MOTION
  (MOVING_TOWARDS_FOE ?x ?t)
  =>
  (at ROOT (assert (MOBILE ?x))))  /*a global fact*/
```

The directive "at ROOT" places the new fact at the root of the interpretation graph, i.e., it is inherited by all existing interpretations.

Other observations depend upon the facts that are currently true in a "world" and, therefore, may have only local consequences inside particular interpretations. For example, if two image features *A* and *B* lie on opposite sides of the Fuzzy FOE and they are getting closer to each other, then they must be in relative motion in 3D space. If an interpretation exists that considers at least one of the two entities (x,y) stationary, then (at least) the other entity cannot

be stationary (i.e., it must be mobile). The following rule "fires within" each interpretation that considers the first entity (x) stationary:

```
(defrule RELATIVE_MOTION
  (OPPOSITE_FOE ?x ?y ?t) /* first observation */
  (CONVERGING ?x ?y ?t)  /* second observation */
  (STATIONARY ?x) /* true inside an interpretation */
 =>
  (assert (MOBILE ?y)))  /* local to this interpretation */
```

While some image observations allow direct conclusions about motion in the scene, other observations hold cues about the stationary 3D structure. If the *exact* location of the FOE is known, then the depth of each stationary point (i.e., its 3D distance from the camera) is proportional to the rate of expansion (from the FOE) of its image (Equation 7). Consequently, for the Fuzzy FOE, where a set of potential FOE locations is given, the distance Z(*A*) of a stationary point *A* is determined as an interval instead of one single number:

$$\mathbf{Z}^{\min}(A) \leq \mathbf{Z}(A) \leq \mathbf{Z}^{\max}(A) .$$

Therefore, a point *A* is closer in 3D than another point *B*, if the corresponding ranges of depth do not overlap, i.e.,

$$\mathbf{Z}^{\max}(A) < \mathbf{Z}^{\min}(B) \rightarrow (CLOSER\ A\ B) .$$

Since this conclusion only holds if both features are actually stationary, the following rule fires only within a suitable interpretation (if it exists):

```
(defrule CLOSER_FROM_EXPANSION
  (STATIONARY ?x)        /*interpretation where */
  (STATIONARY ?y)        /*both are stationary */
  (< (Zmax ?x) (Zmin ?y))  /*no overlap in range */
 =>
  (assert (CLOSER ?x ?y))).
```

To compare the ranges of 3D points, another criterion can be used which does not require the rate of expansion from the FOE. Instead, the change of distances *between* certain pairs of features is observed. If two stationary points lie on the same side of the FOE and the distance between them is becoming smaller, then the *inner* feature (i.e., the one which is nearer to the FOE) is also closer in 3D space. This is a valuable test for features that are relatively near to each other in the image. It can be employed even if the image is not derotated and the location of the FOE is either only known very roughly or is completely outside the field of view (i.e., for a side-looking camera):

```
(defrule CLOSER_FROM_CHANGING_DISTANCE
  (STATIONARY ?x)          /*interpretation where */
  (STATIONARY ?y)          /*both are stationary, */
  (SAME_SIDE_OF_FOE ?x ?y) /*both on the right,  */
  (CONVERGING ?x ?y)       /*dist. is shrinking  */
  (INSIDE ?x ?y)           /*x is nearer to FOE  */
 =>
  (CLOSER ?x ?y).
```

Verification Rules

While the purpose of the generation rules is to establish new hypotheses and conclusions, the purpose of *verification rules* is to review interpretations after they have been created and, if possible, prove that they are false. When a hypothesis is found to be inconsistent with some new observation, it is usually removed from the QSM. Any interpretation that is based on such a hypothesis is removed simultaneously. Since we are always trying to come up with a single (and hopefully correct) scene interpretation, this mechanism is important for pruning the search tree.

Verification rules are typically based on image observations that, used as generators, would produce a large number of unnecessary conclusions. For example, the general layout of the scene seen from the top of a land-based vehicle suggest the rule of thumb that things which are *lower* in the image are generally closer to the camera. Although this rule is not strong enough to draw direct conclusions, it may be used to verify existing hypotheses:

```
(defrule LOWER_IS_CLOSER_HEURISTIC
  (CLOSER ?x ?y) (BELOW_THE_HORIZON ?x ?t)
  (BELOW_THE_HORIZON ?y ?t) (BELOW ?y ?x ?t)
 =>
  /*mark this interpretation as conflicting*/
  (assert (CONFLICT LOWER/CLOSER ?x ?y))).
```

Whenever an existing hypothesis (CLOSER ?x ?y) violates the above rule of thumb, this rule fires and marks the interpretation as conflicting. How the conflict is eventually resolved depends upon the global state of the QSM. Simply removing the afflicted interpretation would create an empty model if this interpretation was the only one. This task is handled by a set of dedicated *conflict resolution rules*.[1]

The kind of rules described up to this point are mainly based upon the geometry of the imaging process, i.e., perspective projection. Other important visual clues are available from occlusion analysis, perceptual grouping, and semantic interpretation. *Occlusion* becomes an interesting phenomenon when features of higher dimensionality than points are employed, such as lines and regions. Similarities in form and motion found by *perceptual grouping* allow us to assemble simple features into complex objects. Finally, as an outcome of the recognition process, *semantic* information may help to disambiguate the scene interpretation. If an object has been recognized as a building, for example, it makes every interpretation obsolete that considers this object mobile. For all these various lines of reasoning, the QSM serves as a common platform.

Meta Rules

In summary, the construction of the QSM and the search for the most plausible scene interpretation are guided by the following meta rules:

- Always tend towards the "most stationary" (i.e. most conservative) solution. By default all new entities are considered stationary.
- Assume that an interpretation is feasible unless it can be proved to be false (the principle of "lack of conflict").
- If a new conclusion causes a conflict in one but not in another current interpretation, then remove the conflicting interpretation.
- If a new conclusion cannot be accommodated by any current interpretation, then create a new, feasible interpretation and remove the conflicting ones.

5. EXPERIMENTAL RESULTS USING QSM

5.1 Fuzzy FOE Results

In the following, the results of the FOE-algorithm and computation of the vehicle's velocity over ground are shown on a real image sequence taken from the moving ALV. The original sequence was provided on standard video tape with a

frame-rate of 30 per second. Out of this original sequence, images were taken in 0.5 second intervals, i.e., at a frame rate of 2 per second in order to reduce the amount of storage and computation. The images were digitized to a spatial resolution of 512×512, using only the Y-component (luminance) of the original color signal.

Figure 34 shows the edge images of 16 frames with the points being tracked labeled with ascending numbers. We have developed an adaptive windowing technique as an extension of relaxation labeling disparity analysis for the selection and matching of tracked points.[9] The actual image location of each point is the lower left corner of the corresponding mark. The resulting data structure consisted of a list of point observations for each image (time), e.g.,

time t_0: ($(p_1\ t_0\ x_1\ y_1)$ $(p_2\ t_0\ x_2\ y_2)$ $(p_3\ t_0\ x_3\ y_3)$ ...)

time t_1: ($(p_1\ t_1\ x_1\ y_1)$ $(p_2\ t_1\ x_2\ y_2)$ $(p_3\ t_1\ x_3\ y_3)$ ...)

. . .

Points are given a unique label when they are encountered for the first time. After the tracking of a point has started, its label remains unchanged until this point is no longer tracked. When no correspondence is found in the subsequent frame for a point being tracked, either because of occlusion, or the feature left the field of view, or because it could not be identified, tracking of this point is discontinued. Should the same point reappear again, it is treated as a new item and given a new label. Approximately 25 points per image have been selected in the sequence shown in Figure 34.

In the search for the Focus of Expansion, the optimal FOE-location from the previous pair of frames is taken as the initial guess. For the very first pair of frames (when no previous result is available), the location of the FOE is guessed from the known camera setup relative to the vehicle. The points which are tracked on the two cars (24 and 33) are assumed to be known as moving and are not used as reference points to compute the FOE, vehicle rotation, and velocity. This information is eventually supplied by the reasoning processes in conjunction with the *Qualitative Scene Model*.

Figure 35 shows the results of computing the vehicle's motion for the same sequence as in the previous figure. Each frame t displays the motion estimates for the period between t and the previous frame $t-1$. Therefore, no estimate is available at the first frame (182). Starting from the given initial guess, the FOE-algorithm first searches for the image location, which is not prohibited and where the error function (equation 35) has a minimum.

The optimal horizontal and vertical shift resulting at this FOE-location is used to estimate the vehicle's rotations around the X- and Y-axis. This point, which is the initial guess for the subsequent frame, is marked as a small circle inside the shaded area. The equivalent rotation components are shown graphically on a $\pm$ 1° scale. They are relatively small throughout the sequence such that it was never necessary to apply intermediate derotation and iteration of the FOE-search. Along with the original displacement vectors (solid lines), the vectors obtained after derotation are shown with dashed lines.

After the location with minimum error has been found, it is used as the seed for growing a region of potential FOE-locations. The growth of the region is limited by two restrictions:

- The ratio of maximum to minimum error inside the region is limited, i.e., $E_n^i/E_n^{min} = \rho^i \leq \rho^{lim}$ (see equation 40 for the definition of the error function E_n). No FOE-location for which the error ratio ρ^i exceeds the limit ρ^{lim} is joined to the region. Thus the final size of the region depends on the shape of the error function. In this example, the ratio ρ^{lim} was set at 4.0. Similarly, no *prohibited* locations (Figure 25) are considered.
- The maximum size of the region M is given. he given FOE-region. region regardless of their error values. The resulting error ratio $\rho^{max} = \max(\rho^i)$ for the points inside the region indicates the shape of the error function for this area. A low value for the ratio ρ^{max} indicates a flat error function. The value for ρ^{max} is shown as *FOE-RATIO* in every image.

For the computation of absolute vehicle velocity, only a few prominent displacement vectors were selected in each frame pair. The criterion was that the vectors are located below the FOE and their length is more than 20 pixels. The endpoints of the selected (derotated) vectors are marked with dark dots. The parameter used for the computation of absolute advancement is the height of the camera above the ground, which is 3.3 meters (11 feet).

5.2 Motion Detection and Tracking

Following the computation of the FOE locations in each of the frames in the sequence, the QSM processes the images and determines the motion of the moving objects and builds a 3D representation of the environment as described in section 4. Figures 36 (a-f) show the complete scene interpretations starting at frame 183 up to frame 197. Interpretations are ranked by their number of stationary entities, i.e., "Interpretation 1" is ranked higher than "Interpretation 2" if both exist. During this run, the maximum number of concurrent interpretations was two. Whenever two interpretations exist at the same time, they are lined-up horizontally in Figure 36. Otherwise, interpretations are displaced to indicate that they refer to different points in time. Entities are marked as stationary or mobile. Entities which carry no mark (just the label) are stationary and have not been found to be closer than any other entity in the scene. A square without a pointer in any direction means that this entity is considered mobile, but that the direction of movement could not be determined for the current frame interval.

The scene contains two moving objects, a car (24) which has passed the ALV and is moving away throughout the sequence and another vehicle (33), approaching the ALV on the same road, which appears in frame 185.

After the first pair of frames (frame 183), two interpretations are created due to the movement of point 24 (the receding car). Interpretation 1 is preferred because it contains 23 stationary entities instead of 18 in interpretation 2. The latter interpretation is discarded due to inconsistent expansion of the points considered moving downwards.

A single interpretation is pursued from frame 184 until frame 194. In this period, no object motion other than the one caused by point 24 is observed. However, the perception of the 3D structure of the stationary part of the scene is continuously refined by adding new *closer*-relationships between entities. Point 24 is always considered mobile, although the direction of its movement cannot be identified between every pair of frames.

After frame 195, two interpretations again become feasible, this time caused by the movement of the approaching car (point 33). Again, the (correct) alternative 1 was ranked higher due to the larger number of stationary entities.

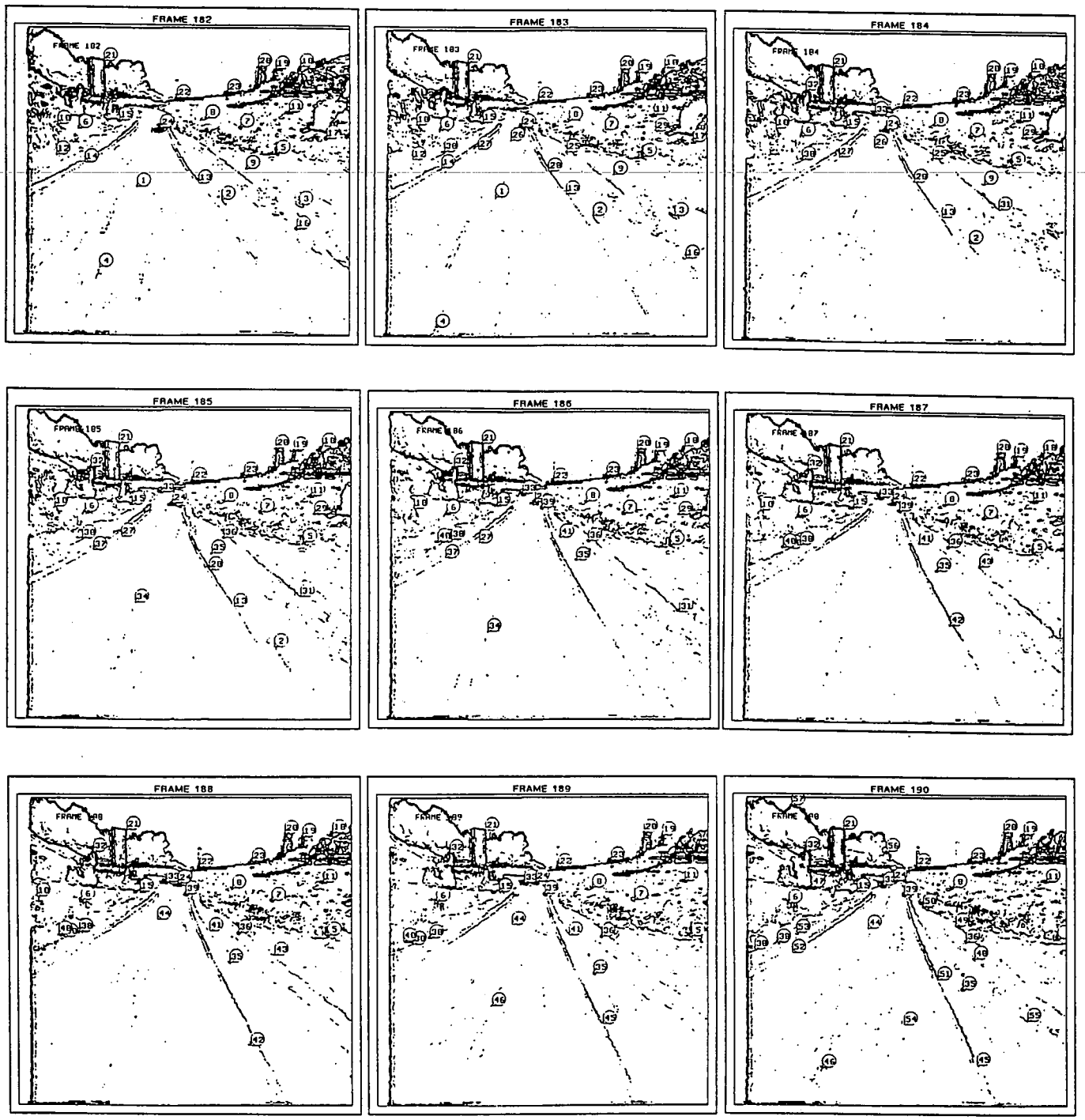


Figure 34: Original image sequence taken from the moving ALV after edge detection and point selection. The selected points are located at the lower-left corners of their marks. *(a)* Frames 182-190 of the original image sequence. The scene contains two moving objects, one car moving away from the ALV (point 24) and another car approaching the ALV (point 33).

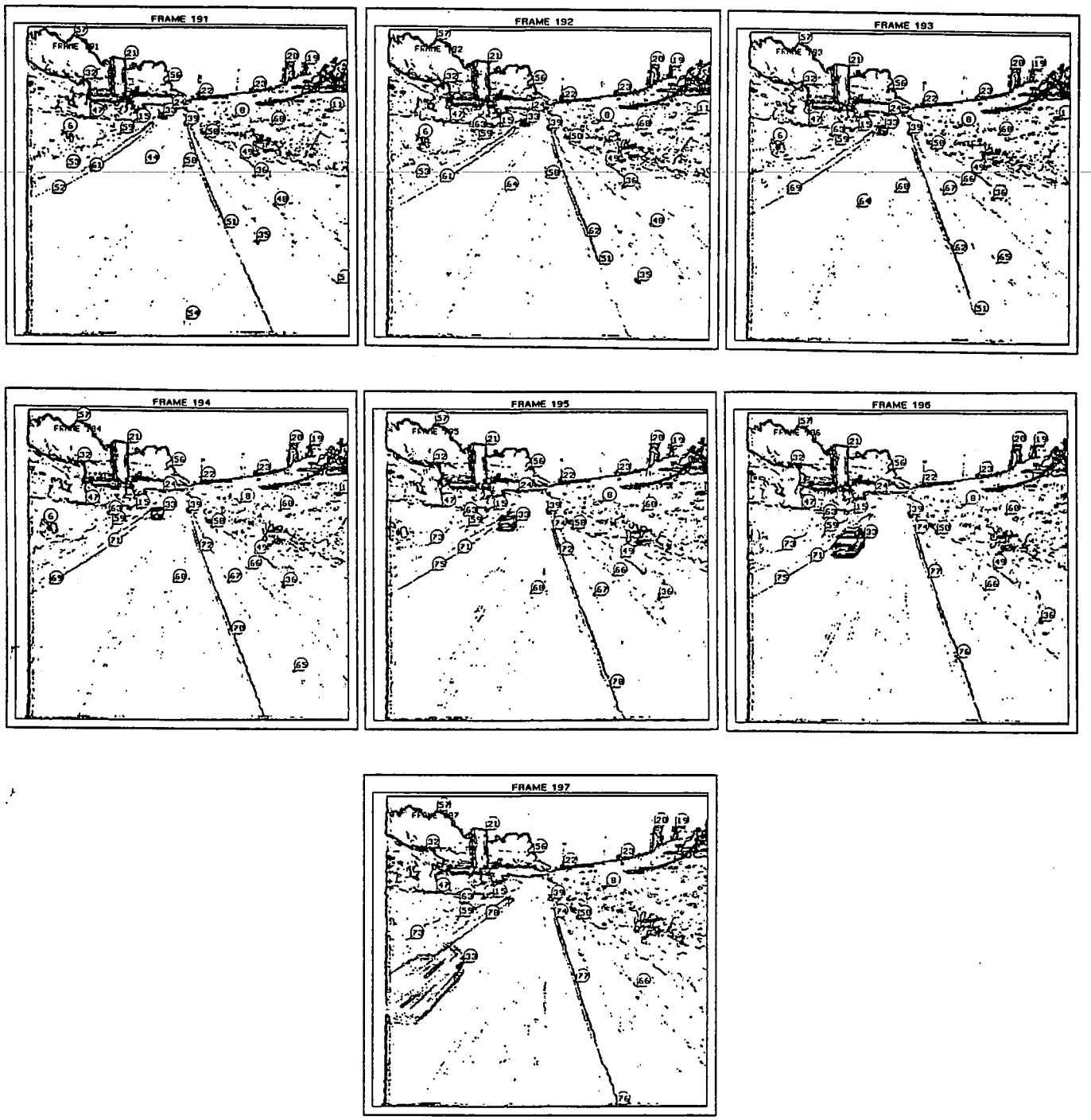


Figure 34(b): Frames 191-197 of the original image sequence after edge detection and point selection.

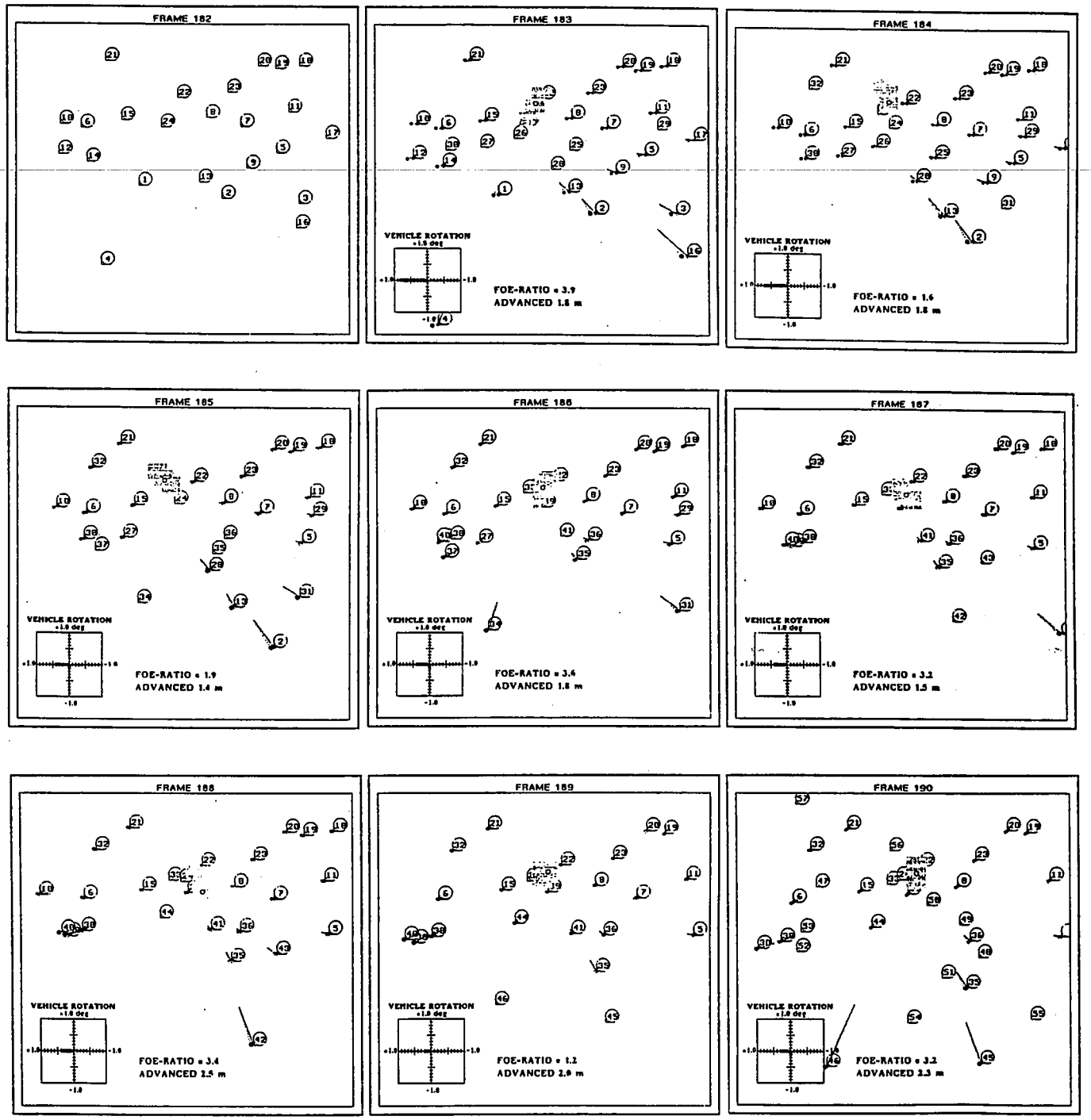


Figure 35: Displacement vectors and estimates of vehicle motion for the image sequence shown in Figure 34. The shaded area marks the possible FOE locations, the circle inside is the FOE with the lowest error value. *FOE-RATIO* measures the flatness of the error function inside this area. The absolute advancement of the vehicle is estimated in meters, vehicle rotation is plotted in a coordinate grid over ±1.0°. *(a)* Displacement vectors and estimates of vehicle motion for frames 182-190 shown in Figure 34(a).

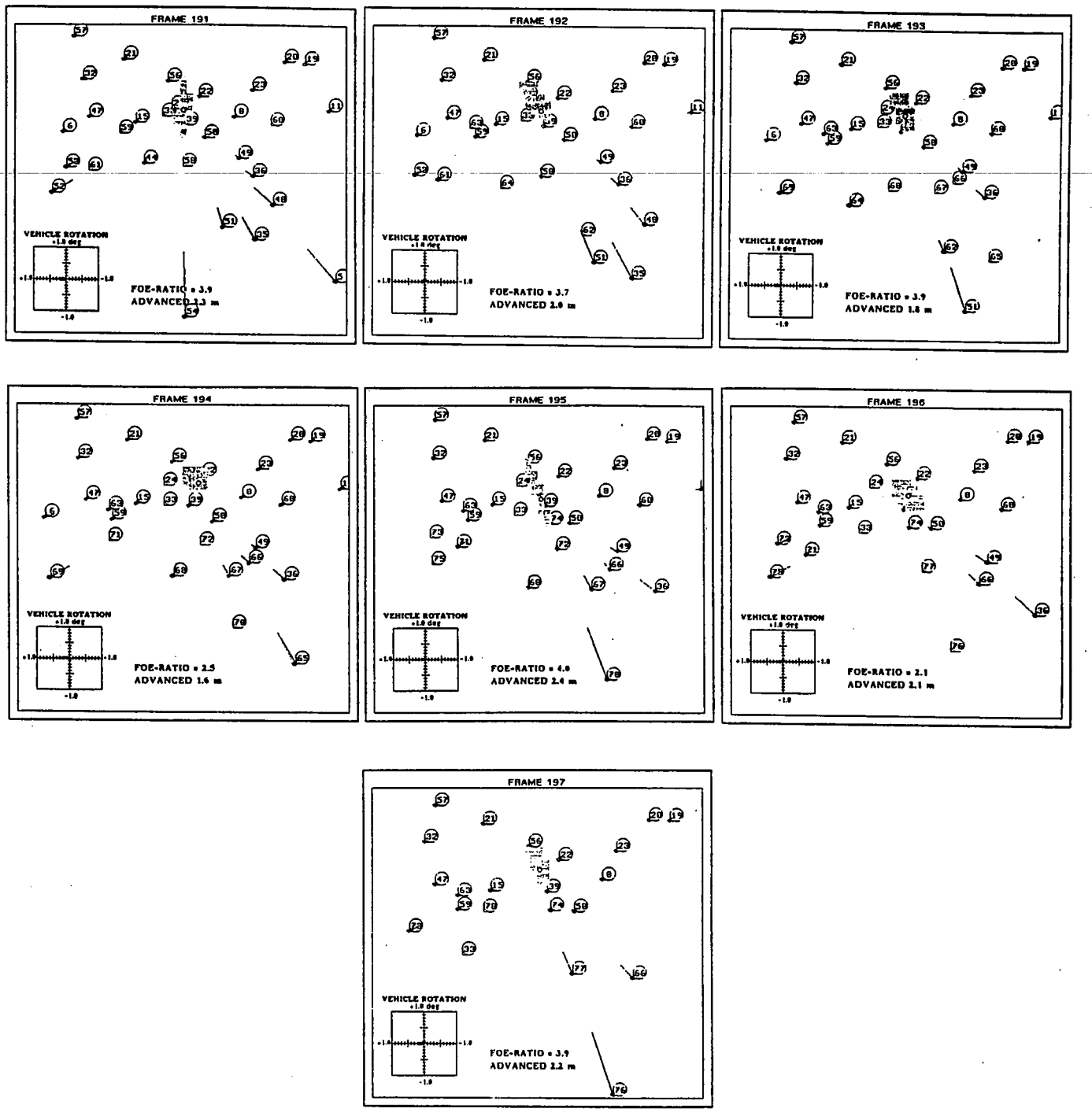


Figure 35(b): Displacement vectors and estimates of vehicle motion for frames 191-197 shown in Figure 34(b).

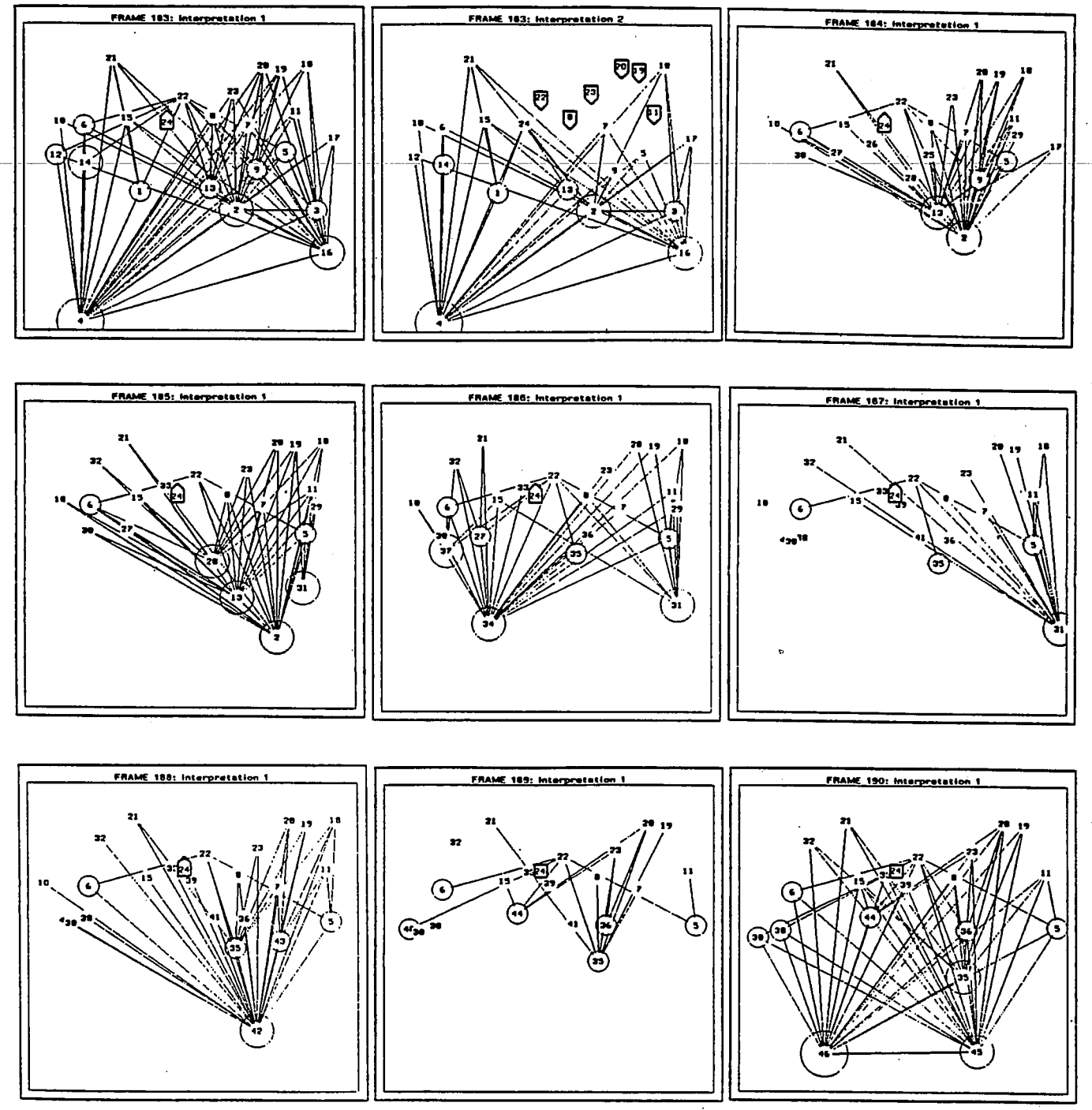


Figure 36: Scene interpretations for image sequence in Figure 34. *(a)* Frames 183-190. After the first pair of frames two interpretations are created due to the movement of point 24 (the receding car). Interpretation 1 is ranked higher because it contains 23 stationary entities instead of 18 in interpretation 2. The latter interpretation is discarded after frame 184 due to inconsistent expansion of the points considered moving downwards. The single interpretation from frame 184 is pursued, because no object motion other than the one caused by point 24 is observed in this period.

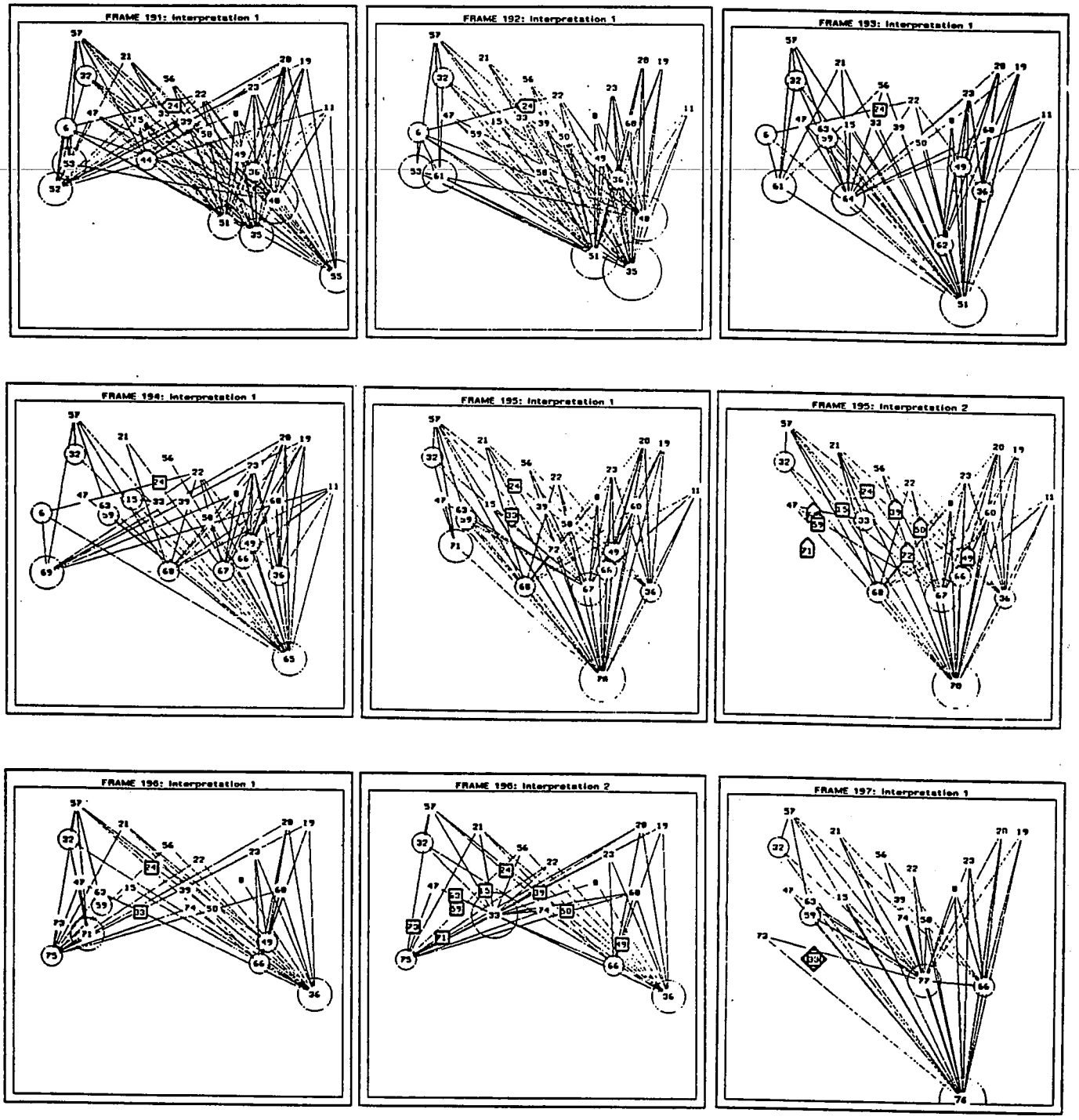


Figure 36: *(b)* Frames 191-197. The 3D structure of the stationary part of the scene is continuously refined by adding new *closer*-relationships between entities.

The two interpretations are pursued simultaneously until frame 197. At this time, a pending *closer*-conflict between point 33 and 76 in interpretation 2 is resolved. Point 76 is clearly closer to the ALV because it is near the bottom of the image, but the faster expansion of point 33 contradicts that. Therefore, as in the synthetic example, interpretation 2 can be discarded in favor of interpretation 1 and point 33 is correctly identified as approaching the ALV (Figure 36(f)).

This experiment shows that the *Qualitative Scene Model* is robustly maintained under real-world conditions, i.e., noise, distortion, imperfect derotation, and location of the FOE. The number of simultaneous interpretations is quite small (maximum is 2) and the correct interpretation clearly ranks higher at any point in time.

6. CONCLUSIONS

In this paper, we presented a qualitative approach to scene understanding for mobile robots in dynamic environments. The challenge of understanding such image sequences is that stationary objects do not appear as stationary in the image and mobile objects do not necessarily appear to be in motion. Consequently, the detection of 3D motion often requires reasoning far beyond simple 2D change analysis.

The approach taken here clearly departs from related work by following a strategy of qualitative, rather than quantitative, reasoning and modeling. All the numerical efforts are packed into the computation of the Focus of Expansion (FOE), which is is accomplished entirely in 2D. To cope with the problems of noise and errors in the displacement field, we determine a region of possible FOE-locations instead of a single FOE. Termed the *Fuzzy FOE*, it is probably one of the most robust techniques of this kind available today.

We showed on sequence of data that, even without knowing the exact location of the FOE, powerful conclusions about motion and 3D scene structure are possible. From these clues, we construct and maintain an internal 3D representation, termed the *Qualitative Scene Model*, in a generate-and-test cycle over extended image sequences. This model also serves as a platform for other visual processes, such as occlusion analysis, perceptual grouping, and object recognition. To overcome the ambiguities inherent to dynamic scene analysis, multiple interpretations of the scene are pursued simultaneously.

The examples given in the paper show the fundamental operation of our approach on real images produced by the Autonomous Land Vehicle. We also wanted to demonstrate that some apparently simple situations require relatively complex paths of reasoning. Of course, the exclusive use of displacement vectors from point features is a limiting factor. To exploit a larger part of the information contained in the image and to demonstrate the full potential of our approach, more complex 2D features, such as lines and regions, will be employed in our future work.

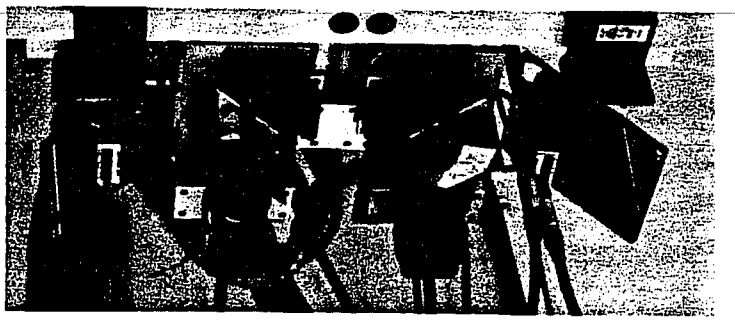
REFERENCES

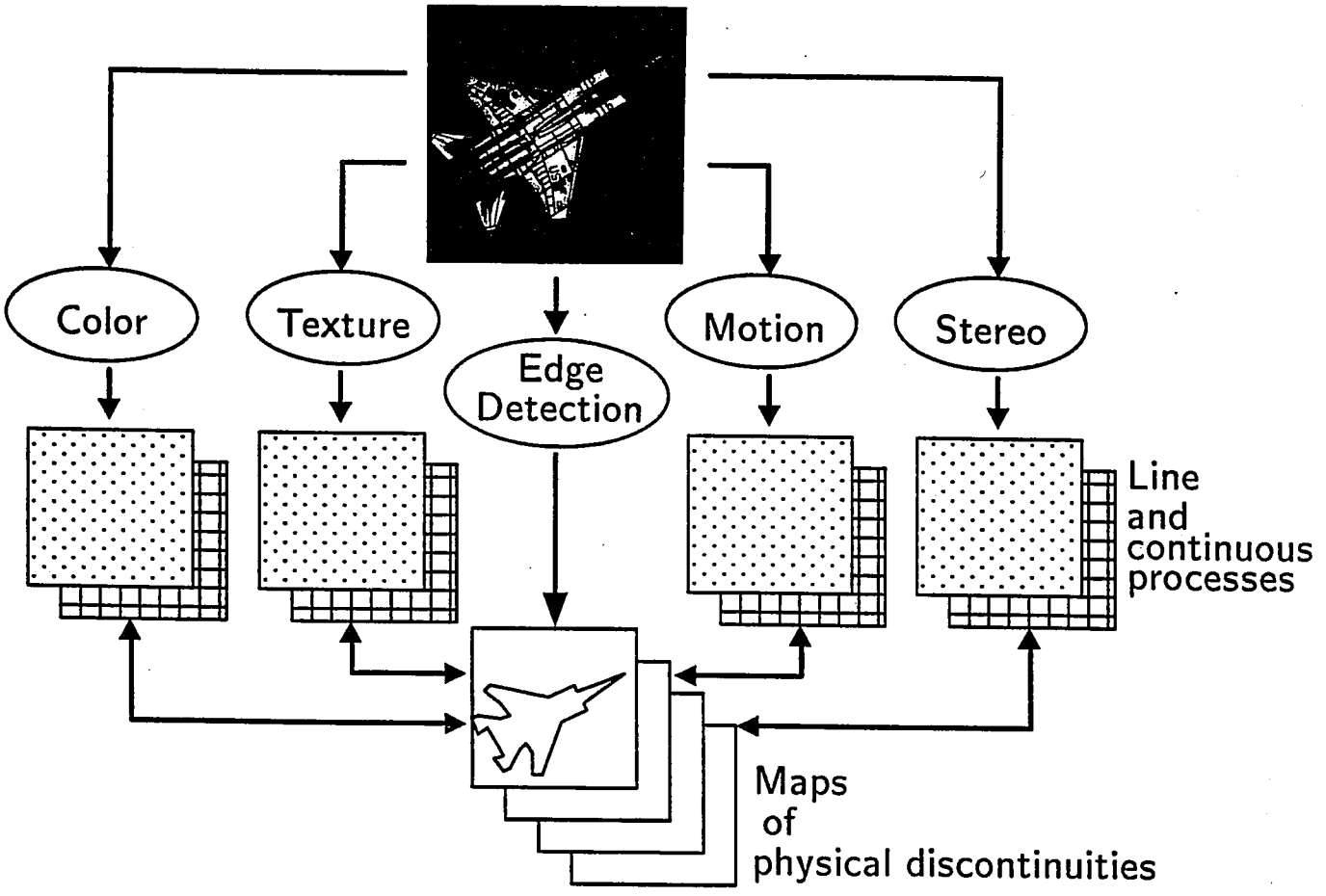
1. B. Bhanu and W. Burger, "DRIVE: Dynamic Reasoning from Integrated Visual Evidence," *Proc. DARPA Image Understanding Workshop*, pp. 581-588 Morgan Kaufmann Publishers, (Feb. 1987).
2. B. Bhanu and W. Burger, "Approximation of Displacement Field Using Wavefront Region Growing," *Computer Vision, Graphics and Image Processing*, (March 1988).
3. S. Bharwani, E. Riseman, and A. Hanson, "Refinement Of Environmental Depth Maps Over Multiple Frames," Proc. IEEE Workshop on Motion, Kiawah Island Resort, pp. 73-80 (May 1986).
4. R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York (1973).
5. O.D. Faugeras, F. Lustman, and G. Toscani, "Motion and Structure from Point and Line Matches," Proc. of 1st International Conference on Computer Vision, pp. 25-34, London (June 1987).
6. R.M. Haralick, "Using Perspective Transformations in Scene Analysis," *Computer Graphics and Image Processing* **13** pp. 191-221 (1980).
7. E.C. Hildreth and N.M. Grzywacz, "The Incremental Recovery of Structure from Motion: Position vs. Velocity Based Formulations," Proc. IEEE Workshop on Motion, Kiawah Island Resort, pp. 137-143 (May 1986).
8. R. Jain, "Direct Computation of the Focus of Expansion," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-5**(1) pp. 58-64 (January 1983).
9. J. Kim and B. Bhanu, "Motion Disparity Analysis Using Adaptive Windows," Technical Report 87SRC38, Honeywell Systems & Research Center (June 1987).
10. H.P. Moravec, "Towards Automatic Visual Obstacle Avoidance," Proc. 5th International Joint Conference on Artificial Intelligence, pp. 584 (August 1977).
11. H.-H. Nagel, "Image Sequences - Ten (octal) Years - From Phenomenology towards a Theoretical Foundation," Proc. Intern. Conf. on Pattern Recognition, Paris, pp. 1174-1185 (1986).
12. K. Prazdny, "Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinear Moving Observer," *Computer Graphics and Image Processing* **17** pp. 238-248 (1981).
13. K. Prazdny, "On the Information in Optical Flows," *Computer Vision, Graphics, and Image Processing* **22** pp. 239-259 (1983).
14. D. Regan, K. Beverly, and M. Cynader, "The Visual Perception of Motion in Depth," *Scientific American*, pp. 136-151 (July 1979).
15. J.H. Rieger, "Information in Optical Flows Induced by Curved Paths of Observation," *J. Opt. Soc. Am.* **73**(3) pp. 339-344 (March 1983).
16. W.B. Thompson and J.K. Kearney, "Inexact Vision," Proc. IEEE Workshop on Motion, Kiawah Island Resort, pp. 15-21 (1986).
17. R.Y. Tsai and T.S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**(1) pp. 13-27 (January 1984).
18. S. Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambridge, Mass. (1979).
19. A. Verri and T. Poggio, "Qualitative Information in the Optical Flow," Proc. DARPA Image Understanding Workshop, Los Angeles, pp. 825-834 (February 1987).

Volume I

PROCEEDINGS:

Image Understanding Workshop





Sponsored by:

Defense Advanced Research Projects Agency
Information Science and Technology Office

DARPA

SC

April 1988

Image Understanding Workshop

Proceedings of a Workshop
Held at
Cambridge, Massachusetts

April 6-8, 1988

Volume I

Sponsored by:

Defense Advanced Research Projects Agency
Information Science and Technology Office

This document contains copies of reports prepared for the DARPA Image Understanding Workshop. Included are results from both the basic and strategic computing programs within DARPA/ISTO sponsored projects.

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

The views and conclusions contained in this document are those of the authors and should not be interpreted as necesarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.