On Computing a 'Fuzzy' Focus of Expansion for Autonomous Navigation*

W. BURGER+ and B. BHANU

Honeywell Systems & Research Center
3660 Technology Dr, Minneapolis MN 55418, U.S.A.

Abstract: The *Focus of Expansion* (FOE) is an important concept in dynamic scene analysis, particularly where translational motion is dominant, such as in mobile robot applications. In practice, it is difficult to determine the *exact* location of the FOE from a given set of displacement vectors due to the effects of camera rotation, digitization, and noise. Instead of a single image location, we propose to compute a connected region, termed 'Fuzzy' FOE, that marks the approximate direction of camera heading. The 'Fuzzy' FOE provides numerous clues about the 3-D scene structure and independent object motion. In this paper we discuss the main problems of the classic FOE approach and concentrate on the details of computing the 'Fuzzy' FOE for a camera undergoing translation and rotation in 3-D space. We present results on real outdoor images.

1. Introduction

The FOE approach relies upon the well-known fact that, under forward camera translation, all image features seem to diverge radially from a particular image location, called the *focus of expansion* (FOE), which marks the direction of vehicle heading [7, 11, 14]. The 3-D distance of stationary environmental features can subsequently be obtained by measuring how fast their images diverge away from the FOE. Thus the computation of camera motion is achieved in a purely two-dimensional step which is completely separated from reconstructing the scene's 3-D structure.

In practice, locating the FOE accurately is not a trivial task for arbitrary camera motion and especially difficult under noisy conditions. Some researchers have therefore assumed planar motion or even pure camera translation [2, 5, 6, 10, 12]. Our solution to the problem is not to search for a *single* FOE-location but to compute a two-dimensional *FOE-region*, which we call the 'Fuzzy' FOE. Despite its apparent lack of precision, the 'Fuzzy' FOE can be employed as a powerful tool in dynamic scene analysis as we have shown elsewhere [1, 4]. It naturally supports a qualitative style of reasoning that receives growing interest within the vision community [3, 16, 17]. This paper concentrates upon the computational aspects of the 'Fuzzy' FOE.

In the following section we discuss two basic FOE techniques: *FOE-from-Rotation* and *Rotation-from-FOE*. Although the latter approach turns out to be superior, the actual location of the FOE can generally not be determined under noisy conditions. In Section 3 we extend the *Rotation-from-FOE* approach to the 'Fuzzy' FOE. A description of the actual algorithm is followed by some typical results on real image sequences which were taken from the Autonomous Land Vehicle (ALV).

2. Searching for a Single FOE

When a camera performs pure translation along a straight line in space, the images of all stationary features seem to diverge from a single location commonly called the *focus of expansion* (FOE). In reality, however, the vehicle not only translates but also rotates more or less about its three major axes. The movement **M** of a land vehicle between two points in time can be sufficiently approximated by a translation **T** followed by rotations about the horizontal axis $\mathbf{R}_\theta$ (panning) and the vertical axis $\mathbf{R}_\varphi$ (tilting). Any 3-D point X = (x,y,z) in the camera-centered co-ordinate frame is transferred by the camera movement **M** to a new location X'=(x',y',z') by

$$\mathbf{M}:\ X \rightarrow X' = \mathbf{R}_\varphi\, \mathbf{R}_\theta\, (\mathbf{T} + X), \tag{1}$$

where

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

If the observed scene is completely stationary, the effects of the camera movement **M** upon the image can be described by a 2-D transformation **d** (for *displacement*), which takes the original image **I** to the following image **I'**. The 3-D rotations $\mathbf{R}_\varphi$ and $\mathbf{R}_\theta$ and the translation **T** have their equivalents in **d** as three separate (non-linear) 2-D operations $\mathbf{r}_\varphi$, $\mathbf{r}_\theta$, and $\mathbf{t}_S$:

$$\mathbf{d}:\ \mathbf{I} \rightarrow \mathbf{I}' = \mathbf{r}_\varphi\, \mathbf{r}_\theta\, \mathbf{t}_S\, \mathbf{I}\,. \tag{2}$$

Since pure camera rotations do not create new aspects of the environment, the corresponding 2-D transformations $\mathbf{r}_\varphi$ and $\mathbf{r}_\theta$ are effectively mappings of the image onto itself. Thus the effects of camera rotations can be undone when the rotation angles are known. Conversely, the image effects $\mathbf{t}_S$ of pure camera translation depend upon each 3-D point's actual location in space, i.e., the 3-D structure of the scene. The subscript S (for *structure*) in $\mathbf{t}_S$ indicates this dependence.

For the purpose of clarity we introduce a (hypothetical) intermediate image **I*** which is the result of the pure camera translation **T**, i.e.

$$\mathbf{t}_S:\ \mathbf{I} \rightarrow \mathbf{I}^*\,. \tag{3}$$

* This work was supported by the Defense Advanced Research Projects Agency under contract DACA 76-86-C-0017 and monitored by the U.S. Army Engineer Topographic Laboratories.

+ Current address: Johannes Kepler University, Department of Systems Sciences, A-4045 Linz, Austria.

Notice that the image $\mathbf{I}^*$ is never really observed, except in the special case of pure camera translation, i.e. when $\mathbf{I}^*=\mathbf{I}'$ (Fig.1). However, $\mathbf{I}^*$ has two important properties: *First*, all displacement vectors between corresponding points in $\mathbf{I}$ and $\mathbf{I}^*$ seem to diverge from a particular image location (x_f,y_f) (i.e. the FOE), unless the camera does not translate at all. We call a displacement field with this property a *radial mapping* from $\mathbf{I}$ to $\mathbf{I}^*$. *Second*, for given pan and tilt angles φ and θ, $\mathbf{I}^*$ can be obtained regardless of the 3-D scene structure by applying the *inverse rotational mappings* $\mathbf{r}_\varphi^{-1}$ and $\mathbf{r}_\theta^{-1}$ (which always exist) to the observed image $\mathbf{I}'$:

$$\mathbf{I}^* = \mathbf{r}_\theta^{-1}\,\mathbf{r}_\varphi^{-1}\,\mathbf{I}' = \mathbf{t}_S\,\mathbf{I}\,. \qquad (4)$$

Several methods have been suggested for extracting the required displacement vector fields (e.g., [9, 13, 15]). The task of computing the motion parameters can be stated as a search problem in the space of φ, θ and $\mathbf{t}$. The relationship in (4) suggests two different search strategies for separating the motion components:

(1) FOE from Rotation: Successively apply combinations of inverse rotational mappings $\mathbf{r}_{\theta 1}^{-1}\,\mathbf{r}_{\varphi 1}^{-1}$, ..., $\mathbf{r}_{\theta m}^{-1}\,\mathbf{r}_{\varphi m}^{-1}$ to the second image $\mathbf{I}'$, until the resulting image $\mathbf{I}^*$ is a radial mapping with respect to the original image $\mathbf{I}$. Then locate the FOE $\mathbf{x}_{fm}$ in image $\mathbf{I}$.

(2) Rotation from FOE: Successively select FOE-locations $\mathbf{x}_{f1}$, $\mathbf{x}_{f2}$,..., $\mathbf{x}_{fn}$ in the original image $\mathbf{I}$ and then determine the inverse rotational mappings $\mathbf{r}_{\theta n}^{-1}\,\mathbf{r}_{\varphi n}^{-1}$ that yield a radial mapping with respect to the final FOE $\mathbf{x}_{fn}$.

Both alternatives were investigated in the course of this work under the conditions of restricted but realistic camera motion. As it turns out, the major problem with the *FOE-from-Rotation* approach is to determine, whether a given mapping of image points is radial or (more important) *how close* it is to being radial when the location of the FOE is still unknown.

2.1 FOE from Rotation

In this method, the image motion is decomposed in two steps. First, the rotational components are hypothesized and their inverses are applied to the second image $\mathbf{I}'$, thus partially *derotating* the image. If the assumed rotations are accurate, the resulting displacement field diverges from a single image location. The second step verifies that the derotated displacement field is actually radial and then determines the location of the FOE.

There are several ways to obtain an initial estimate for the camera's rotation. If the platform has sufficient inertia, the results from one pair of frames can be carried over to the following pair to serve as the initial guess. Another version is to use features known to be at far distance as a reference for camera rotation. Since the amount of displacement caused by translation depends upon the 3-D distance of the corresponding environmental features, points at far distance are not significantly affected by translation. Thus all the image displacement of these points is due to camera rotations, which can be easily estimated from individual displacement vectors.

After applying a particular derotation to the second image $\mathbf{I}'$ (to remove the effects of the hypothesized rotations), the question is how much the resulting displacement field deviates from the desired *radial* mapping. Prazdny [14] estimates the disturbance of the displacement field from the variance of the intersections of one (extended) displacement vector with all other vectors. We evaluated two similar approaches, one using the intersections of displacement vectors with imaginary horizontal and vertical lines and another based on *Linear Correlation* [1]. These results indicate a general problem associated with methods that rely upon *extending and intersecting* displacement vectors. By extending displacement vectors, existing errors (e.g. from digitization and noise) are multiplied. Consequently, these approaches are unreliable when the amount of camera translation is small and necessarily fail when the camera does not translate at all.

2.2 Rotation from FOE

This approach avoids the problem of extending displacement vectors by starting with a hypothesized FOE location. Once a particular FOE $\mathbf{x}_f$ has been selected, the problem is to find the rotational mappings $\mathbf{r}_\varphi^{-1}$ and $\mathbf{r}_\theta^{-1}$ that, when applied to the image $\mathbf{I}'$, will result in an optimal radial mapping with respect to $\mathbf{I}$ and $\mathbf{x}_f$. The error measure chosen for this purpose uses the deviation of the displacement vectors from straight rays originating from the hypothesized FOE. Given a set of corresponding image points $\{(\mathbf{x}_i,\mathbf{x}_i') \mid \mathbf{x}_i\in \mathbf{I},\ \mathbf{x}_i'\in \mathbf{I}'\}$ and some FOE-location $\mathbf{x}_f$, the error is defined as the inner product between the vectors $(\mathbf{x}_f\rightarrow\mathbf{x}_i)$ and $(\mathbf{x}_f\rightarrow\mathbf{x}_i')$:

$$E = \sum_i \left[\frac{1}{|\mathbf{x}_i-\mathbf{x}_f|}\,(\mathbf{x}_i-\mathbf{x}_f)\times(\mathbf{x}_i'-\mathbf{x}_f) \right]^2. \qquad (5)$$

Under perspective transformation, image points move along hyperbolic paths when the camera performs pure rotation, i.e., the resulting displacement is not uniform over the entire image plane. However, if the amount of rotation is small or the focal length of the camera is large, we can replace the nonlinear mappings $\mathbf{r}_\varphi^{-1}$ and $\mathbf{r}_\theta^{-1}$ by a linear shift vector $\mathbf{s}_{\varphi,\theta}=(s_x,s_y)$ which is independent of the image location [1]:

$$\mathbf{I}^* = \mathbf{r}_\theta^{-1}\,\mathbf{r}_\varphi^{-1}\,\mathbf{I}' \approx \mathbf{s}_{\varphi,\theta} + \mathbf{I}'. \qquad (6)$$

With $\mathbf{s}_{\varphi,\theta}$ as a free variable, the error measure (5) becomes

$$E = \sum_i \frac{1}{|\mathbf{x}_i-\mathbf{x}_f|^2}\left[\,(\mathbf{x}_i-\mathbf{x}_f)\times(\mathbf{x}_i'-\mathbf{x}_f+\mathbf{s}_{\varphi,\theta})\right]^2 \qquad (7)$$

This second-order error function can be minimized with standard numerical techniques to obtain an optimal value for $\mathbf{s}_{\varphi,\theta}$. From the resulting shift vector $\mathbf{s}_{\varphi,\theta}$, the corresponding (approximate) rotation angles φ and θ are easily found. If the resulting rotation exceeds the limits imposed by the assumption of small rotation, the image may be first derotated by those angles using the exact (non-uniform) mapping, and then processed again, until the residual rotations are sufficiently small. Thus even large amounts of camera rotation can be handled accurately without violating the given assumptions. Under usual conditions no more than a single iteration is necessary.

The following procedure[1] `Test_Single_FOE` examines *one* hypothetical FOE-location $\mathbf{x}_f$ in a given pair of images $\mathbf{I}$ and $\mathbf{I}'$. It uses the functions `Optimal_Rotation` for computing the optimal shift vector $\mathbf{s}_{\varphi,\theta}$ and the error value for the resulting

[1] The pseudocode notation used for the algorithms allows functions to return multiple values (as in CommonLISP), e.g. `(x,y,z):= PROC(a,b)` as well as multiple assignments, e.g. `(u,v):= (r,s)`.

displacement field (by minimizing Equ.7). Function `Derotate_Image` effectively undoes the effects of a camera rotation by φ and θ. The constant angle ρ_{max} marks the limit of *small* camera rotation (about 4°):

```
Test_Single_FOE(x_f, I, I´):
 /* Compute the camera rotation and resulting error
 given the two images I,I´ for the hypothesized
 FOE x_f.*/
 I*:= I´;
 repeat                    /*usually only 1 iteration*/
    (φ,θ,error):= Optimal_Rotation(x_f, I, I*);
    I*:= Derotate_Image(I*,φ,θ);
    until (|φ| ≤ ρ_max) & (|θ| ≤ ρ_max)
 return (I*,φ,θ,error).
```

Fig.2 shows the resulting error values at selected FOE-locations around the actual FOE for a sparse displacement field simulating 2° of horizontal camera rotation. The size of the circle at each location indicates the amount of error (i.e. the deviation from the radial displacement field) that would result if *that* location were picked as the FOE. The smooth shape of this error function makes it possible that, even with a poor initial guess, the global optimum can be found by local search methods.

An important criterion is the function's behavior when the amount of camera translation is small or the displacement vectors are noisy. Fig.3 shows the effects of varying the average length of the displacement vectors in the range of 4-60 pixels. It is obvious that longer displacement vectors result in a sharper minimum around the actual FOE. The effects of noise are demonstrated in Fig.4, using the same displacement vectors as in Fig.2. Apparently, the error function flattens out with increasing levels of noise, although the generic shape of the function does not change. However, the location of minimum error may move considerably away from the actual FOE. These are the major motivations for computing the 'Fuzzy' FOE as described in the following section.

3. The 'Fuzzy' FOE

Essentially, the local shape of the error function in Figures 2-4 reflects how well the FOE is actually specified by the given displacement field. The goal of the 'Fuzzy' FOE is to somehow explore this shape to construct a connected image region that contains the *real* FOE with high certainty. The way we have chosen to compute the 'Fuzzy' FOE is to (1) search for a location $\mathbf{x}_{min}$ with local minimum error starting from some initial location $\mathbf{x}_0$ and then (2) *grow* a connected region around $\mathbf{x}_{min}$ until some criterion is met. In particular, the region is complete when the accumulated *error volume* exceeds a predefined limit. Intuitively, the 2-D error function could be seen as a bowl into which we pour a certain quantity of liquid (error volume) and then take the boundary of the resulting surface as the 'Fuzzy' FOE. The following algorithm `Compute_Fuzzy_FOE` and the subsequent auxiliary procedures explain these steps:

```
Compute_Fuzzy_FOE(I, I´, max_err):
 x_0:= Guess_Initial_FOE;
 x_min:= Search_Min_FOE(x_0, I, I´);
 Fuzzy_FOE:= Grow_Fuzzy_FOE(x_min, I, I´, max_err);
 return (Fuzzy_FOE).
```

```
Search_Min_FOE(x_0, I, I´):
 x_min:= x_0
 (φ_min, θ_min, err_min):= Test_Single_FOE(x_min, I, I´);
 repeat
    (x_n, φ_n, θ_n, err_n):= Min_Neighbor({x_min}, I, I´);
    if (err_n < err_min) then
       (x_min, φ_min, θ_min, err_min):= (x_n, φ_n, θ_n, err_n);
    until (err_n ≥ err_min);
 return (x_min).
```

```
Grow_Fuzzy_FOE(x_min, I, I´, max_err);
 Fuzzy_FOE:= {x_min}; /* initialize FOE set */
 err_max:= err_min; /* max error found so far */
 err_vol:= 0;
 repeat
    (x_n, φ_n, θ_n, err_n):= Min_Neighbor(Fuzzy_FOE, I, I´);
    Fuzzy_FOE:= Fuzzy_FOE ∪ {x_n};
    Δerr:= err_n - err_max;  err_max:= err_n;
    err_vol:= err_vol + Δerr * card(Fuzzy_FOE);
    until (err_vol > max_err);
 return (Fuzzy_FOE).
```

The auxiliary function `Min_Neighbor` examines all neighbors of a given set of image locations and returns the location with minimum error value:

```
Min_Neighbor(FOE_Region, I, I´):
 err_min:= ∞;
 forall x_i ∈ FOE_Region
    forall x_a: (Neighbors_p(x_a, x_i) & x_a ∉ FOE_Set)
       (φ_a, θ_a, err_a):= Test_Single_FOE(x_a, I, I´);
       if (err_a < err_min) then
       (x_min, φ_min, θ_min, err_min):= (x_a, φ_a, θ_a, err_a);
       endfor
    endfor
 return (x_min, φ_min, θ_min, err_min).
```

The standard 8-neighborhood may be used for defining the set of image locations adjacent to one pixel (predicate `Neighbors_p`). However, to improve the efficiency of the search procedure, we actually select the neighbors on a grid of varying size. The resolution of this grid is successively altered from coarse to fine during the initial search for the optimal FOE.

Fig.5 demonstrates the result of applying this algorithm to a real image sequence taken from the moving ALV. One of the original images, after edge detection, is shown in Fig.5a. In this example, the displacement vectors were obtained by hand-tracking point features in the edge images. Still, the end points of the displacement vectors are not very accurate and thus the amount of noise is considerable. The shaded area in Fig.5b marks the resulting 'Fuzzy' FOE, with a small circle inside at the location of minimum error. The jagged outline of the 'Fuzzy' FOE is caused by the relatively coarse grid (10 pixels) used for growing the region. The coordinate frame in the lower left-hand corner displays the estimated amount of vehicle rotation (φ and θ) about the horizontal and vertical axes. For this frame pair, θ is approximately 0.5°, while φ is close to zero. Since the time elapsed between the given frame pair is about 0.2 seconds, the ALV is currently turning towards the left at a rate of 2.5° per second. The *derotated* displacement field, shown with dotted vectors, is almost perfectly radial.

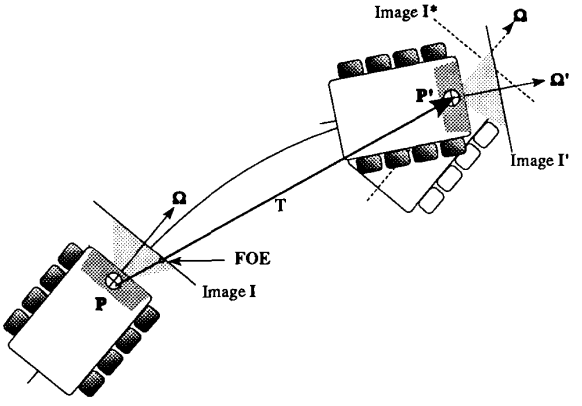


Figure 1. Vehicle translation and rotation as separate steps. The vehicle first translates from position **P** to position **P**′and then changes its orientation from Ω to Ω′. The intermediate image **I*** would be observed after the translation **T**. The FOE is located where the 3-D vector **T** intersects image **I**.

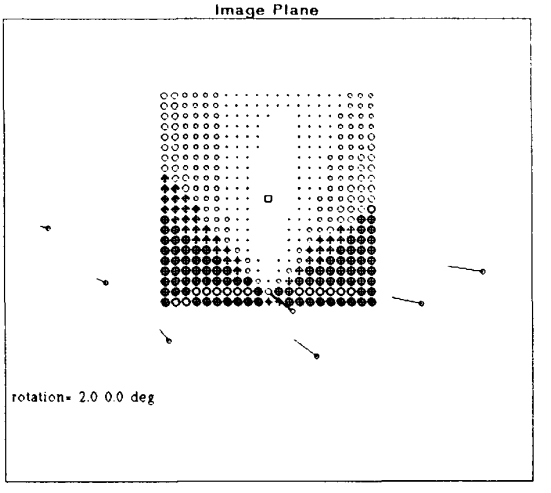
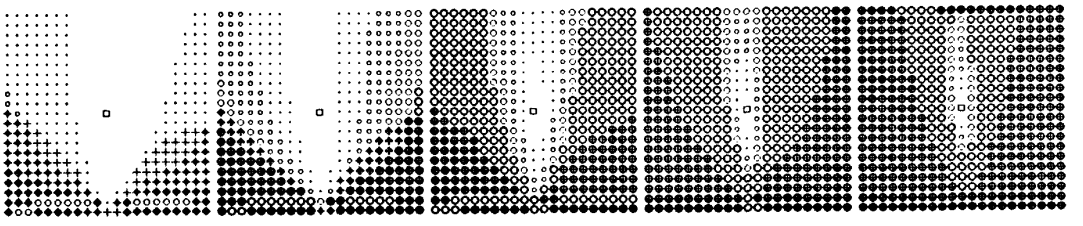
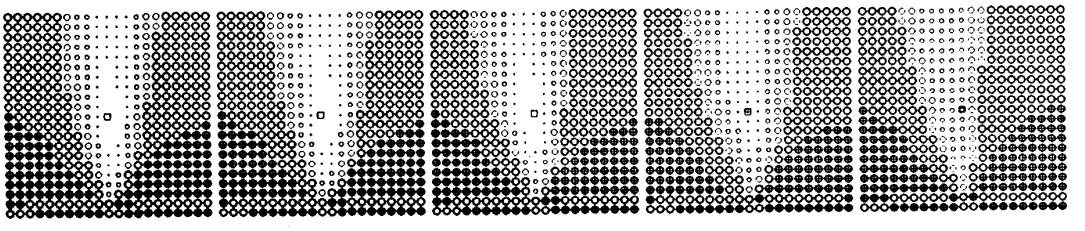


Figure 2. Displacement vectors and resulting error values for selected FOE locations. The diameter of each circle indicates the amount of error (minimizing Equ.5) for that particular FOE location. The error function is plotted over an area of ±100 pixels centered around the actual FOE (marked by a small square).



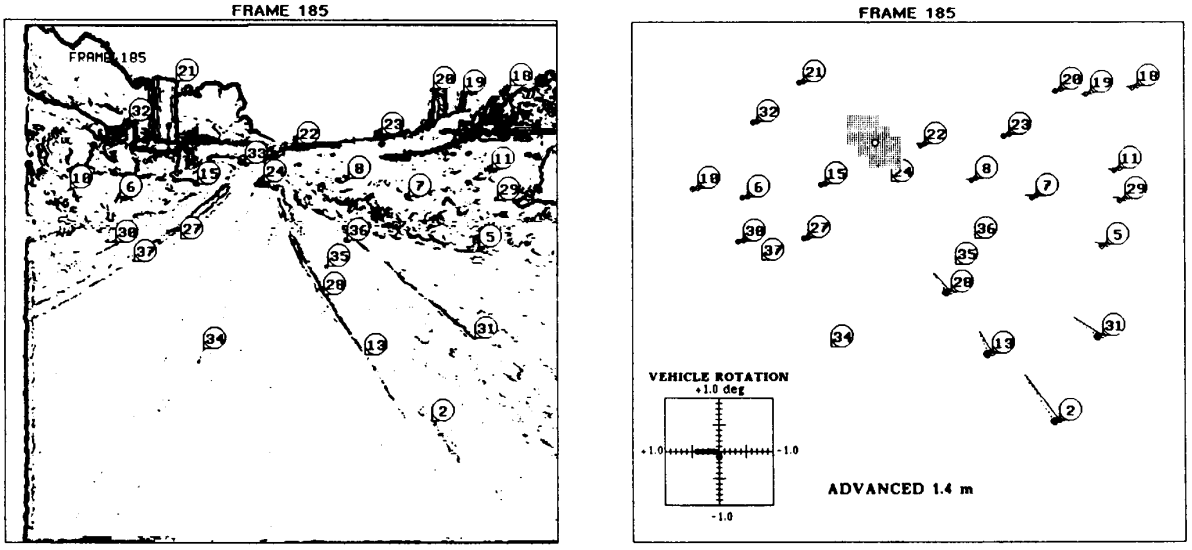
(a) *(b)* *(c)* *(d)* *(e)*

Figure 3. Effects of average length of displacement vectors upon the shape of the error function. The same displacement field as in Fig.2 was used. *(a)* Average length is 4 pixels, *(b)* 8 pixels, *(c)* 20 pixels, *(d)* 40 pixels, *(e)* 60 pixels. Obviously, the FOE is better defined by longer displacement vectors.



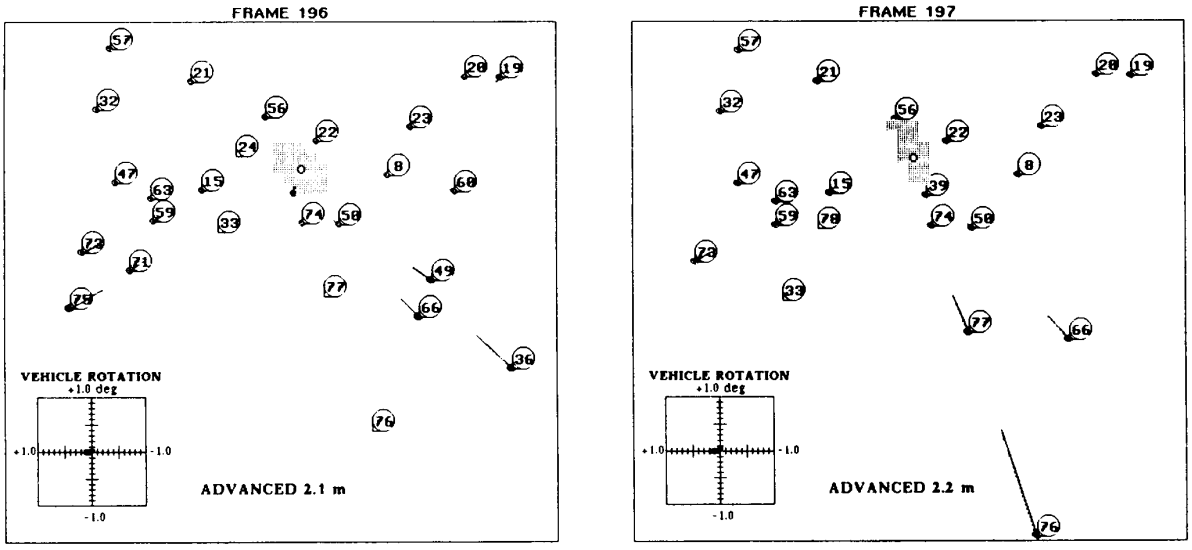
(a) *(b)* *(c)* *(d)* *(e)*

Figure 4. Effects of uniform noise upon the shape of the error function. The same displacement field as in Fig.2 was used. *(a)* Zero noise, *(b)* ±0.5 pixels, *(c)* ±1.0 pixels, *(d)* ±2.0 pixels, *(e)* ±4.0 pixels. The error function becomes flat with increasing noise levels.



(a) *(b)*

Figure 5. Result of computing the Fuzzy FOE on a real image sequence taken from the moving ALV. *(a)* Original image after edge detection. *(b)* The shaded region marks the resulting Fuzzy FOE with a small circle for the location of minimum error (i.e., the optimal FOE). Estimated vehicle rotation (φ, θ) is plotted in a co-ordinate frame in the range of $\pm 1.0°$.



(a) *(b)*

Figure 6. Relationship between displacement field structure and Fuzzy FOE. The group of dominant (long) displacement vectors in the lower right-hand corner of both images *(a)* and *(b)* results in elongated FOE regions.

Two additional examples from the same sequence are shown in Fig.6 to demonstrate the relationship between the structure of the given displacement field and the resulting shape of the 'Fuzzy' FOE. For some reason, the dominant displacement vectors are found in the lower right-hand corner of both frames (Figures 6a and 6b). Since the error measure (Equ.5) implicitly assigns more weight to longer displacement vectors, those dominant vectors have a direct influence upon the shape of the FOE. This explains the elongated shape of the 'Fuzzy' FOE in the two images. For typical road scenes, where long displacement vectors are generally located in the lower parts of the image, the horizontal position of the FOE is therefore better defined (by a small diameter) than its vertical position.

If the displacement vectors are not obtained by manual tracing (as in the examples presented here) but computed automatically, there is, in addition to the noise problem, a potential danger of erroneous point matches. Consequently, any FOE algorithm based on displacement data must be sufficiently robust to tolerate those errors. In a recent experiment [8], the 'Fuzzy' FOE algorithm was successfully applied to 262 frames of 5 different (ALV) image sequences with *automatic* selection and matching of point features. Also, the algorithm has been applied with good results to helicopter flight scenes in a fully automatic process.

4. Conclusion

In this paper we presented the 'Fuzzy' FOE as a new robust tool for dynamic scene analysis. Like other FOE techniques, it is particularly suited for motion sequences that exhibit a significant translation component, i.e., for most vehicular applications. In contrast to other methods, however, the 'Fuzzy' FOE performs well even under the conditions of small rotation and noise or at least provides a measure for the quality of the result. Experiments show that even erroneous point matches in the given image displacement vector field can be tolerated, which is a necessity in a fully automated process.

The information obtained from the 'Fuzzy' FOE is useful for a variety of tasks. First, the parameters of self-motion (i.e., direction of translation and rotations) themselves are important for efficient short-term control of the vehicle, such as steering, accelerating, and braking. From the derotated displacement vector field, 3-D layout of the environment can be recovered to support navigation, obstacle avoidance, and manipulation. Essentially, the fact that the FOE is not obtained precisely does not necessarily imply a disadvantage. Instead, the reasoning process becomes more realistic, because the significance of any conclusion based on the 'Fuzzy' FOE can be made explicit. Details on using the 'Fuzzy' FOE for detecting object motion and estimating 3-D scene structure can be found in [1].

Although we have only considered point features for the computation of image displacement vectors so far, there is good reason to believe that more complex 2-D features would result in more reliable frame-to-frame matches. Lines, regions, and contour segments appear to be good candidates to enhance the overall performance of this approach.

REFERENCES

1. Bhanu B., Burger W., "Qualitative Motion Detection and Tracking of Targets from a Mobile Platform," Proc. DARPA Image Understanding Workshop, Cambridge Mass., pp. 289-318 (April 1988).
2. Bolles R.C., Baker H.H., "Epipolar-Plane Analysis: A Technique for Analyzing Motion Sequences," Proc. IEEE Workshop on Motion: Representation and Control, pp. 168-178 (Oct.1985).
3. Burger W., Bhanu B., "Qualitative Motion Understanding," Proc. 10th Int. Joint Conf. on Artificial Intelligence IJCAI'87, pp. 819-821 (Aug. 1987).
4. Burger W., Bhanu B., "Dynamic Scene Understanding for Autonomous Mobile Robots," Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR'88, Ann Arbor, Michigan (June 1988).
5. Dutta R., Manmatha R., Riseman E.M., Snyder M.A., "Issues in Extracting Motion Parameters and Depth from Approximate Translational Motion," Proc. Darpa IU Workshop, Cambridge Mass., pp. 945-960 (April 1988).
6. Jain R., "Direct Computation of the Focus of Expansion," *IEEE Trans. Pattern Analysis and Machine Intelligence* **PAMI-5** (1), pp. 58-64 (Jan.83).
7. Jerian C., Jain R., "Determining Motion Parameters for Scenes with Translation and Rotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6** (4), pp. 523-530 (July 1984).
8. Kim.J., Bhanu B., "Motion Disparity Analysis Using Adaptive Windows," Technical Report 87SRC38, Honeywell Systems & Research Center, Minneapolis MN (June 1987).
9. Kories R., Zimmermann G., "A Versatile Method for the Estimation of Displacement Vector Fields from Image Sequences," Proc. IEEE Workshop on Motion: Representation and Analysis, pp. 101-106 (May 1986).
10. Lawton D.T., "Processing Translational Motion Sequences," *Computer Vision, Graphics, and Image Processing* **22**, pp. 114-116 (1983).
11. Longuet-Higgins H.C., Prazdny K., "The Interpretation of a Moving Retinal Image," *Proc.R.Soc.London* B **208**, pp. 385-397 (1980).
12. Marimont D.H., "Projective Duality and the Analysis of Image Sequences," Proc. IEEE Workshop on Motion: Representation and Analysis, pp. 7-14 (May 1986).
13. Nagel H.-H., "Displacement Vectors Derived from Second-Order Intensity Variations in Image Sequences," *Computer Vision, Graphics, and Image Processing* **21**, pp. 85-117 (1983).
14. Prazdny K., "Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinear Moving Observer," *Computer Vision, Graphics, and Image Processing* **17**, pp. 238-259 (1981).
15. Thompson W.B., Barnard S.T., "Lower-Level Estimation and Interpretation of Visual Motion," *IEEE Computer* **14** (8), pp. 20-27 (Aug. 1981).
16. Thompson W.B., Kearney J.K., "Inexact Vision," Proc. IEEE Workshop on Motion, Kiawah Island Resort, pp. 15-21 (May 1986).
17. Verri A., Poggio T., "Qualitative Information in the Optical Flow," Proc. DARPA Image Understanding Workshop, Los Angeles, pp. 825-834 (February 1987).