A Geometric Constraint Method for Estimating 3-D Camera Motion

Wilhelm Burger* and Bir Bhanu
University of California, Riverside

Abstract

We investigate the problem of estimating the camera motion parameters from a 2-D image sequence that has been obtained under combined 3-D camera translation and rotation. Geometrical constraints in 2-D are used to iteratively narrow down regions of possible values in the space of 3-D motion parameters. The approach is based on a new concept called "FOE-feasibility" of an image region, for which an efficient algorithm has been implemented. Results are shown on real images.

1 Introduction

The problem of determining the motion parameters of a moving camera relative to its environment from a sequence of images is important for the application of computer vision to mobile robots. 2-D image motion is measured as a set of displacement vectors between corresponding feature points in successive frames.

Under pure translation, the direction of heading in 3-D is found at the "focus of expansion" (FOE), which is the image location where all displacement vectors intersect, i.e., image features corresponding to the stationary points in the scene seem to "expand" radially from the FOE under forward translation. This strong geometric constraint allows to compute the camera motion directly in 2-D, i.e., without computing the 3-D scene structure at the same time [1, 2, 8, 9, 10, 11, 12]. Unfortunately, computing the *precise* location of the FOE is a difficult problem in real image sequences, due to the effects of camera rotations, noisy displacement vectors, and spatial discretization errors [3, 4].

One solution to this problem is the *Fuzzy FOE* approach described in [3], where a connected *region* of possible FOE locations is computed instead of a single FOE point. However, the Fuzzy FOE region is "grown" by sampling and evaluating individual FOE locations and computing the corresponding rotation parameters. The approach we present here is also based on the idea of representing the FOE as a 2-D *region* instead of a point. The new aspect, however, is that this method does not rely on the evaluation of individual FOE points, but always considers compatible *regions* of translation and rotation parameters.

*Currently on leave from Johannes Kepler University, Linz (Austria). The author was supported in part by the Austrian National Science Foundation (FWF) under grant P8496-PHY.

2 Focus of Expansion under Mixed Camera Motion

2.1 Pure Translation

When the camera undergoes pure translation, each displacement vector $\mathbf{d}_i = (\mathbf{x}_i, \mathbf{x}'_i)$ lies on a ray that passes through the FOE $\mathbf{x}_f$, a property we refer to as *radial* :

$$\mathit{Radial}(\mathbf{d}_i, \mathbf{x}_f) \Leftrightarrow \mathbf{x}'_i - \mathbf{x}_i = \lambda_i (\mathbf{x}_i - \mathbf{x}_f), \quad (1)$$

where λ_i is a scalar. Similarly, a displacement vector field $\mathbf{D}$ is called *radial* with respect to $\mathbf{x}_f$ when $\mathit{Radial}(\mathbf{d}_i, \mathbf{x}_f)$ is true for all its displacement vectors, i.e., $\mathit{Radial}(\mathbf{D}, \mathbf{x}_f) \Leftrightarrow \mathit{Radial}(\mathbf{d}_i, \mathbf{x}_f), \forall \mathbf{d}_i \in \mathbf{D}$.

2.2 Pure Rotation

Arbitrary camera rotation can be described by a sequence of rotations about the X, Y, and Z axes, but in the case of a land vehicle the rotation about the Z-axis can be ignored. A camera rotation $\omega = (\phi, \theta)^T$ moves an image point $\mathbf{x} = (x, y)^T$ along a hyperbolic path [5] to a new location $\mathbf{x}' = \rho(\mathbf{x}, \omega)$, where

$$\rho(\mathbf{x}, \omega) = \frac{f}{-x \cos\phi \sin\theta + y \sin\phi + f \cos\phi \cos\theta} \cdot \begin{bmatrix} x \cos\theta + f \sin\theta \\ x \sin\phi \sin\theta + y \cos\phi - f \sin\phi \cos\theta \end{bmatrix}. \quad (2)$$

Since the effects of a camera rotation are independent of the depth of the observed points, the image effects caused by rotation can be *removed* (if the rotation angles are known) by "derotating" each displacement vector $\mathbf{d}_i = (\mathbf{x}_i, \mathbf{x}'_i)$:

$$\mathbf{d}_i \rightarrow \tilde{\mathbf{d}}_i = (\mathbf{x}_i, \tilde{\mathbf{x}}_i) = \left(\mathbf{x}_i, \rho^{-1}(\mathbf{x}'_i, \omega)\right). \quad (3)$$

Similarly, the entire *displacement vector field* $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$ is derotated by

$$\mathbf{D} \rightarrow \tilde{\mathbf{D}} = \rho^{-1}(\mathbf{D}, \omega) = \{\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2, \ldots, \tilde{\mathbf{d}}_N\} \quad (4)$$

Assuming that a displacement vector $\mathbf{d}_i = (\mathbf{x}_i, \mathbf{x}'_i)$ is the result of camera rotations only (i.e., no translation), the corresponding rotation vector ω is unique:

$$\omega = \bar{\omega}(\mathbf{x}_i, \mathbf{x}'_i) = \bar{\omega}(\mathbf{x}_i, \rho(\mathbf{x}'_i, \omega)). \quad (5)$$

Details regarding the function $\bar{\omega}$ can be found in [3].

2.3 Mixed Motion

The motion of the camera is, in general, a combination of translation and rotation. Consequently, the image effects of the pure motion components combine as well. The problem of locating the FOE in a displacement field $\mathbf{D}$ created by combined camera motion can thus be stated as: "find an image location $\mathbf{x}_f$ and the corresponding rotations ω, such that *Radial*$(\rho^{-1}(\mathbf{D}, \omega), \mathbf{x}_f)$".

3 FOE-Feasibility

At this point, we introduce a weaker condition than (1) by requiring that the displacement vectors caused by translation need only intersect a common *region* instead of a common point. Given a convex image region S and a displacement vector field $\mathbf{D}$, the new relation *S-Radial*$(\mathbf{D}, S)$ holds if all displacement vectors in $\mathbf{D}$ intersect the region S:

$$\textit{S-Radial}(\mathbf{D}, S) \Leftrightarrow \mathbf{d}_i \cap S \neq \emptyset, \ \forall \mathbf{d}_i \in \mathbf{D} \quad (6)$$

An example is shown in Figure 1. It contains three observed displacement vectors $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\}$, $\mathbf{d}_i = (\mathbf{x}_i, \mathbf{x}'_i)$, the translational displacement components $\mathbf{D}^+ = \{\mathbf{d}_1^+, \mathbf{d}_2^+, \mathbf{d}_3^+\}$, $\mathbf{d}_i^+ = (\mathbf{x}_i, \mathbf{x}_i^+)$, the true FOE $\mathbf{x}_f$, and a convex region S. The camera rotation performed between $\mathbf{D}$ and $\mathbf{D}^+$ is $\phi = \theta = 1°$. The focal length is equivalent to the length of the image diagonal, which corresponds to what is considered a standard lens in photographic imaging. Since all three vectors $\mathbf{d}_i$ intersect the region S, *S-Radial*$(\mathbf{D}, S)$ holds.

We also define the new relation *FOE-feasible*$(\mathbf{D}, S)$, which hold if camera rotations ω exist which, when applied to $\mathbf{D}$, would make all displacement vectors intersect with S:

$$\textit{FOE-feasible}(\mathbf{D}, S) \Leftrightarrow \exists\omega, \ \textit{S-Radial}(\rho^{-1}(\mathbf{D}, \omega), S) \quad (7)$$

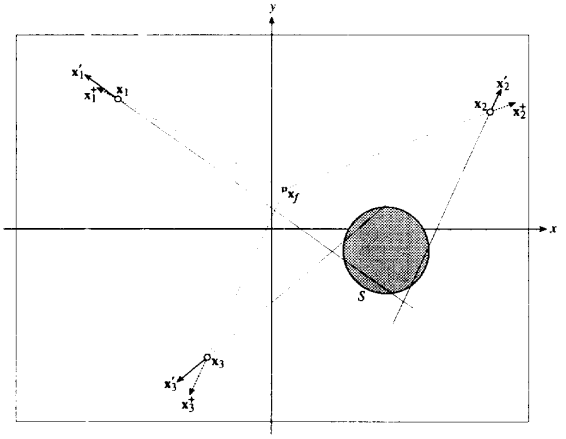


Figure 1: The observed displacement vector field $\mathbf{D}$ consists of three vectors $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$, where $\mathbf{d}_i = (\mathbf{x}_i, \mathbf{x}'_i)$. $\mathbf{D}$ is caused by camera translation (towards $\mathbf{x}_f$) and rotations by $\phi = \theta = 1°$. $\mathbf{D}$ is *S-radial* with respect to the image region S, because all its vectors intersect S, although the true FOE ($\mathbf{x}_f$) is located *outside* of S. $\mathbf{d}_i^+ = (\mathbf{x}_i, \mathbf{x}_i^+)$ are the displacement vectors due to the translation only.

Under the given assumptions (i.e., restricted rotation, stationary environment), *FOE-feasible* $(\mathbf{D}, S)$ is always true if the region S contains the true FOE $\mathbf{x}_f$:

$$\mathbf{x}_f \in S \Rightarrow \textit{FOE-feasible}(\mathbf{D}, S) \quad (8)$$

The inverse, however, does not hold in general, i.e., *FOE-feasible*$(\mathbf{D}, S) \not\Rightarrow \mathbf{x}_f \in S$ (as demonstrated in Figure 1). Thus, finding the FOE by successively *reducing* the size of the region S is not a useful strategy. However, if we can show that a given region is *not FOE-feasible*, then, following (8), S cannot possibly contain the FOE.

3.1 Regions of Motion Parameters

If a given region $\mathbf{S}$ is *FOE-feasible* with respect to a displacement vector field $\mathbf{D}$, then every displacement vector $\mathbf{d}_i \in \mathbf{D}$, after being derotated by some rotation vector $\omega = (\phi, \theta)^T$ (3), must intersect with S.

In general, there is not only a single pair of rotation angles ϕ and θ that would make a displacement vector $\mathbf{d}_i = (\mathbf{x}_i, \mathbf{x}'_i)$ intersect the given region $\mathbf{S}$, but an infinite set of possible value pairs. As shown in Figure 2, any derotation is legal that moves the endpoint $\mathbf{x}'_i$ of the displacement vector into a sector R_i that is bounded by the two tangents of the region S passing though $\mathbf{x}_i$. We denote the set of possible derotation

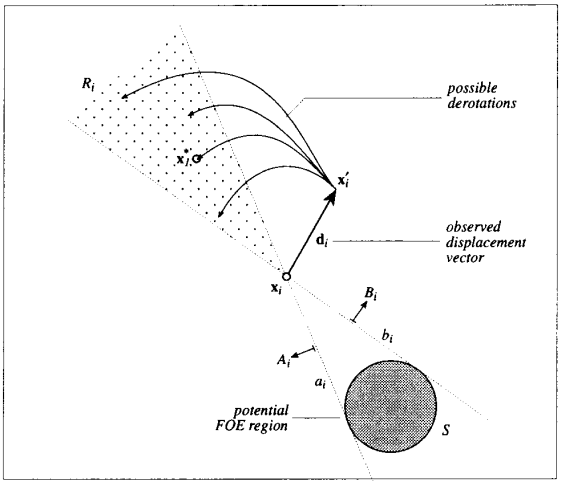


Figure 2: Forcing a single displacement vector to intersect a given image region S. Given a potential FOE region S (which is assumed to contain the true FOE) and a single displacement vector $\mathbf{d}_i$, any derotation that moves the endpoint $\mathbf{x}'_i$ into the sector R_i, defined by the tangents a_i and b_i, makes $\mathbf{d}_i$ *radial* with respect to S. In general, the set of possible rotation angles is infinite.

angles for $\mathbf{d}_i$ with respect to S as $\Omega(\mathbf{d}_i, S)$, where (for $\omega' \in \mathcal{R}^2$):

$$\Omega(\mathbf{d}_i, S) = \{\omega' \mid S\text{-}Radial(\rho^{-1}(\mathbf{d}_i, \omega'), S)\} . \quad (9)$$

In order to instantiate the property *FOE-feasible* $(\mathbf{D}, S)$, we need to find rotation angles $\Omega(\mathbf{D}, S)$ that make *every* displacement vector in $D = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$ intersect S. The set of possible derotation angles is the intersection of the individual sets $\Omega(\mathbf{d}_i, S)$, i.e.,

$$\Omega(\mathbf{D}, S) = \Omega(\mathbf{d}_1, S) \cap \cdots \cap \Omega(\mathbf{d}_N, S) . \quad (10)$$

The set of possible rotation angles $\Omega(\mathbf{d}_i, S)$ corresponds to a connected region in the two-dimensional (ϕ/θ)-space. The connectedness of this region follows from the fact that the corresponding image displacement region R_i (Figure 2) is connected and the rotational mapping function ρ (2) is continuous.

3.2 FOE-Feasibility Algorithm

For computing $\Omega(\mathbf{D}, S)$, we represent the set of possible rotations by a convex polygon in θ/ϕ-space, called the *rotation polygon*. For a given displacement vector field $\mathbf{D}$, $\Omega(\mathbf{D}, S)$ is obtained by iterating over the individual displacement vectors $\mathbf{d}_i$. In each iteration $i = 1 \ldots N$ (N being the number of displacement vectors), the current range of possible rotations is intersected with the possible range of rotations for the displacement vector $\mathbf{d}_i$, i.e.,

$$\Omega^{(i)} \leftarrow \Omega^{(i-1)} \cap \Omega(\mathbf{d}_i, S) , \quad (11)$$

where the initial range $\Omega^{(0)}$ covers the maximum camera rotations, and $\Omega^{(N)} = \Omega(\mathbf{D}, S)$.

Algorithm *FOE-Feasible* $(\mathbf{D}, S)$:

S is the potential FOE region. $\mathbf{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$ is the displacement vector field. $\Omega^{(0)}$ is the region covering the maximum possible camera rotations.

1. $\Omega \leftarrow \Omega^{(0)}$

2. for $i = 1$ to N do :
$\Omega \leftarrow$ *Constrain-Rotations* $(S, \mathbf{d}_i, \Omega)$
if Ω is *nil*, exit
end for

3. If Ω is *nil*, the region S is *not* FOE-feasible. Otherwise Ω is the region of possible camera rotations for the FOE-region S. Return Ω.

□

The key procedure in this algorithm (in Step 2) is defined as follows:

Algorithm *Constrain-Rotations* $(S, \mathbf{d}_i, \Omega^{(i-1)})$:

S is the potential FOE region; $\mathbf{d}_i = (\mathbf{x}_i, \mathbf{x}'_i)$ is the current displacement vector; $\Omega^{(i-1)} = (\omega_1, \ldots, \omega_k)$ is the current rotation polygon.

1. $\Omega^{(i-1)} = (\omega_1, \ldots, \omega_k)$ is mapped to the image space by applying ρ (2) to each of the vertices ω_j: $\Omega^{(i-1)} \mapsto W_i = (\xi_1, \ldots, \xi_k)$, where $\xi_j = \rho(\mathbf{x}'_i, \omega_j)$. The points ξ_j form a polygon $W_i = (\xi_1, \ldots, \xi_k)$ in the image space which approximates the region of possible displacements caused by any rotation $\omega \in \Omega^{(i-1)}$.

2. The polygon W_i is clipped by each of the two half-planes A_i and B_i formed by the tangents on S, a_i and b_i, passing through $\mathbf{x}_i$. The result is a new, possibly empty, polygon in the image domain: $W'_i = (\xi'_1, \ldots, \xi'_l) = W_i \cap A_i \cap B_i$.

3. If W'_i is empty ($l = 0$), then return *nil*. Otherwise, the polygon W'_i is transformed back to the (ϕ/θ)-space to obtain $\Omega^{(i)}$, by applying the function $\bar{\omega}$ (5) to each of its vertices ξ'_j: $W'_i \mapsto \Omega^{(i)} = (\omega'_1, \ldots, \omega'_l)$, where $\omega'_j = \bar{\omega}(\mathbf{x}'_i, \xi'_j)$.

4. Return $\Omega^{(i)}$.

□

This process is illustrated in Figure 3). If no clipping occurs in Step 2 (in the image domain), then the rotation polygon is unchanged, i.e., $\Omega^{(i)} = \Omega^{(i-1)}$. No-

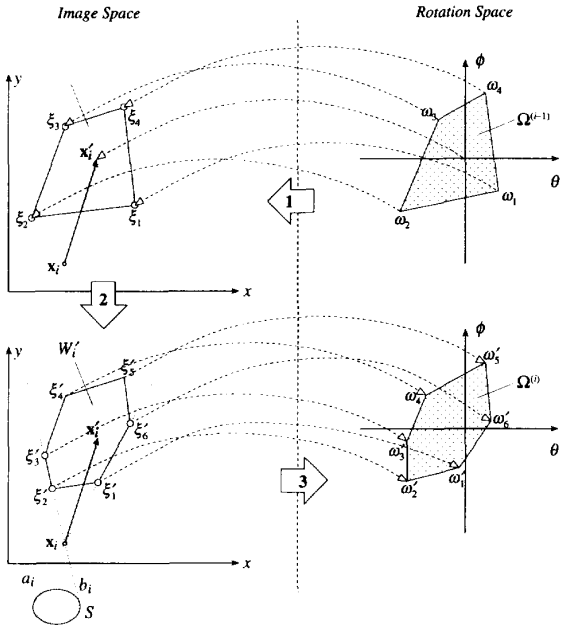


Figure 3: Illustration of the *Constrain-Rotations* algorithm.

tice that the correspondence between the two polygons $\Omega^{(i-1)}$ and W_i is exact at the vertex points, but straight lines in rotation space generally do not map to straight lines in image space [3, 5]. The mapping is exact under orthographic projection, which is equivalent to a perspective projection with an infinite focal length. However, for a reasonable focal length (for f greater than 0.5 times the image diameter, say) and moderate rotation angles (less than $\pm 10°$), the above approximation is sufficient under perspective projection. Of course, the polygonal approximation greatly simplifies the problem of intersecting regions in the image domain (Step 2).

For a given FOE-region S, the function *Constrain-Rotations* is invoked in the *FOE-Feasible* algorithm at most N times, N being the number of displacement vectors in $\mathbf{D}$. *Constrain-Rotations* itself consists of mapping the rotation polygon to the image space and back (Steps 1 and 3), which can be done in linear time with respect to the number of polygon vertices that is proportional to N in the worst case. For intersecting the image polygon with the two half planes in Step 2, we used a variant of the Sutherland-Hodgman algorithm [7], which is also linear in the number of polygon vertices. Thus, as a rough estimate, the worst-case time complexity of *FOE-Feasible* is $\mathcal{O}(N^2)$, although the number of vertices in the rotation polygon is usually much smaller than N and *Constrain-Rotations* terminates as soon as the rotation polygon becomes empty.

4 Locating the FOE

The FOE-feasibility algorithm operates on a given region S that is thought to contain the true FOE, but provides no explicit information how S can be found. As mentioned earlier, even if a region S tests FOE-feasible, the actual FOE may still be located outside of S. On the other hand, S may test negative for one of two reasons: *(a)* The region S does not contain the true FOE. *(b)* The region S *does* contain the FOE, but the displacement vectors are too noisy to make them all intersect S for any camera rotation ω.

While we are not concerned with the first cause here (see [4] for a detailed discussion of separating stationary and moving objects), the other two cases are of importance. The second case was the original reason for using FOE-*regions* instead of points. As a consequence, it is not useful to choose a region S below a certain size, because even if the true FOE were centered within S, the region would never become FOE-feasible.

Therefore, we can use the FOE-feasibility algorithm for locating the FOE or, at least, for constraining the possible location of the FOE. This could be done in several different ways, including the following two basic strategies:

1. Start with a region $S^{(0)}$ of minimum size and enlarge it until *FOE-feasible* $(\mathbf{D}, S^{(k)})$ becomes true. In a typical motion sequence, the direction of camera heading (and thus the location of the FOE) does not change dramatically between successive frames. One way of choosing $S^{(0)}$ is therefore to center $S^{(0)}$ at the FOE computed for the prior frame pair.
2. Start with a region S large enough that it contains the true FOE with certainty and shrink that region until *FOE-feasible*$(\mathbf{D}, S)$ becomes false.

In both cases, the enlarging and shrinking operation, respectively, must maintain the convexity of S. Probably the most effective strategy is a combined search strategy that starts by growing a small seed region, which is subsequently shrunk to constrain the FOE. While S can, in principle, be any convex region, we prefer certain simple shapes over completely arbitrary shapes because of simpler representation and easier computation of the tangents (performed in Step 2 of

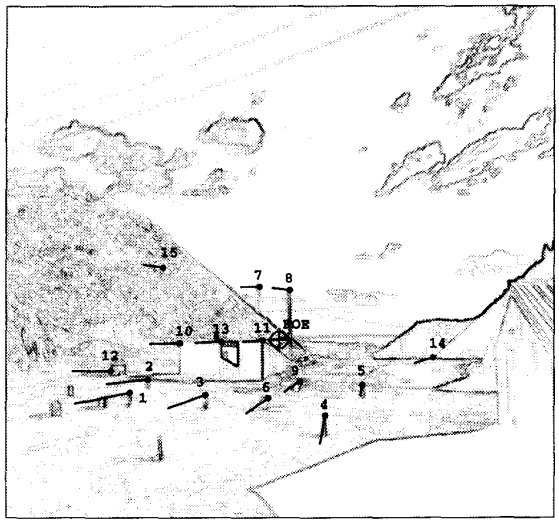


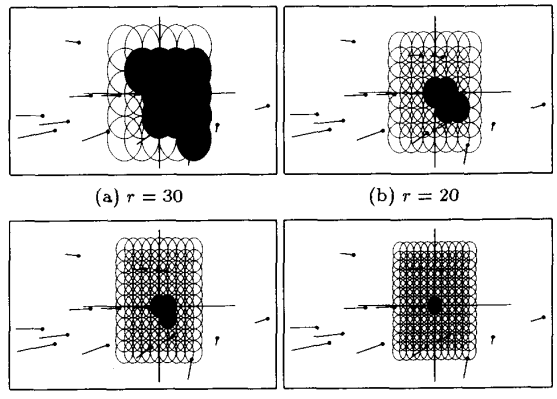
Figure 4: Displacement vectors between frames 1 and 6 of the "Rocket" sequence. The true FOE, computed from ground truth data, is marked $\oplus$. For this frame pair, the ALV moves forward by 4.5 meters and simultaneously rotates about 2.7° to the left.

the *Constrain-Rotations* algorithm). Particularly simple region shapes are rectangles, circles, and ellipses, but one may also consider a hexagonal partitioning of regions.

5 Experiments on Real Data

To demonstrate the approach on real imagery, we used the "Rocket" sequence (Figure 4) compiled by Dutta et al. [6]. For this motion sequence, ground truth information in the form of measured vehicle positions, object positions, and camera parameters is available. The displacement vectors shown in Figure 4 were obtained manually but are inaccurate up to several pixels.

Figure 5 shows the result of evaluating *Foe-Feasible* on an array of circular FOE regions, with the displacement vectors shown in Figure 4. The figure illustrates the fact that the sensitivity of the geometric constraints method increases when the size of the assumed FOE-region is reduced. A large FOE-region will test positive (i.e., the resulting rotation polygon is nonempty) over a larger image area than a smaller FOE-region for the same displacement vector field. Above a certain size, an FOE-region may test positive in all parts of the image, regardless of the true FOE location. At the other extreme, the FOE-region cannot be made



(a) $r = 30$ (b) $r = 20$

(c) $r = 15$ (d) $r = 12$

Figure 5: Probing FOE-feasibility over an array of circular FOE regions around the true FOE. The radius of the regions is varied from 30 down to 12 pixels. Feasible regions are marked black; the true location of the FOE is marked by a cross.

arbitrarily small, because noise would not allow all displacement vectors to intersect that region, even when centered at the true location of the FOE. The region sizes used in Figure 5 approximately cover the practical range for the given displacement vector field. The minimum size of any candidate region depends upon the average length of the translational displacement components (which depends on the amount of camera translation and the distance of the observed points) and the accuracy of the displacement measurements.

Figure 6 illustrates the evolution of the rotation polygon for the single feasible FOE-region of Figure 5d. The rotation polygon is successively constrained as individual displacement vectors are processed. Starting out with a range of $\pm 10°$ for each rotation axis, the rotations are quickly constrained to a very small region.

6 Conclusions

The main motivation for developing this approach to camera motion estimation are the difficulties involved in locating the focus of expansion in real image sequences. While the strong geometrical constraints associated with the FOE make its concept attractive, noise and feature location errors make it hard to deal with in reality. The approach presented here relaxes the FOE localization problem by using the notion of *regions* in both the image space and the rotation space for estimating the motion parameters. Polygons are

used to represent the regions in both domains, which offer a compact description and facilitate efficient processing methods.

References

[1] S. Bharwani, E. Riseman, and A. Hanson. Refinement of environmental depth maps over multiple frames. In *Proc. IEEE Workshop on Motion*, pages 73–80, May 1986.

[2] R.C. Bolles and H.H. Baker. Epipolar-plane analysis: A technique for analyzing motion sequences. In *Proc. IEEE Workshop on Computer Vision*, pages 168–178, October 1985.

[3] W. Burger and B. Bhanu. Estimating 3-D egomotion from perspective image sequences. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(11):1040–1058, 1990.

[4] W. Burger and B. Bhanu. *Qualitative Motion Understanding.* Kluwer Academic Publishers, 1992.

[5] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[6] R. Dutta, R. Manmatha, L.R. Williams, and E.M. Riseman. A data set for quantitative motion analysis. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 159–164, 1989.

[7] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, 2nd edition, 1992.

[8] R. Jain. Direct computation of the focus of expansion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5:58–64, January 1983.

[9] D.N. Lee. The optic flow field: The foundation of vision. *Phil. Trans. R. Soc. Lond. B*, 290:169–179, 1980.

[10] H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. R. Soc. Lond. B*, 208:385–397, 1980.

[11] K. Prazdny. Determining the instantaneous direction of motion from optical flow generated by a curvilinear moving observer. *Computer Graphics and Image Processing*, 17:238–248, 1981.

[12] J.H. Rieger. Information in optical flows induced by curved paths of observation. *Journal of the Optical Society of America*, 73(3):339–344, March 1983.

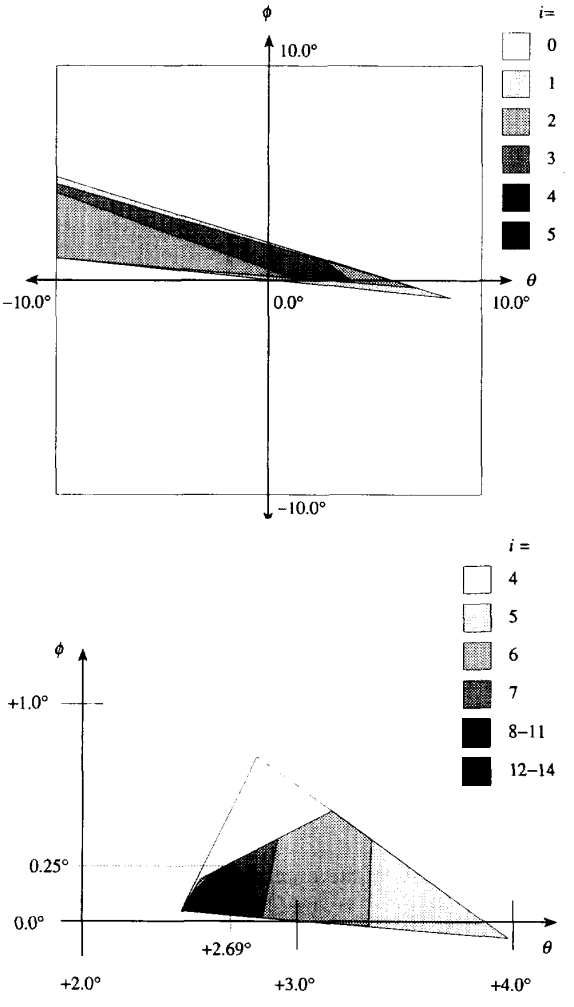


Figure 6: Evolution of the rotation polygon for the FOE-region shown in Figure (d). Initially, for $i = 0$, the range of possible rotations is $\pm 10°$ (*top*). As individual displacement vectors are processed ($i = 1 \ldots 14$), the range of possible camera rotations is successively constrained to a very small region (*bottom*), which shows an enlarged section of the rotation space. Notice that some displacement vectors (9, 10, 11, 13, 14) leave the rotation polygon unchanged. The rotation angles computed from the ground truth data ($\phi = 0.25°, \theta = 2.69°$) lie inside the final rotation polygon.