International Journal of Computer Vision, 1, 7–55 (1987)

Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion*

ROBERT C. BOLLES, H. HARLYN BAKER, AND DAVID H. MARIMONT
Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025

Abstract

We present a technique for building a three-dimensional description of a static scene from a dense sequence of images. These images are taken in such rapid succession that they form a solid block of data in which the temporal continuity from image to image is approximately equal to the spatial continuity in an individual image. The technique utilizes knowledge of the camera motion to form and analyze slices of this solid. These slices directly encode not only the three-dimensional positions of objects, but also such spatiotemporal events as the occlusion of one object by another. For straight-line camera motions, these slices have a simple linear structure that makes them easier to analyze. The analysis computes the three-dimensional positions of object features, marks occlusion boundaries on the objects, and builds a three-dimensional map of "free space." In our article, we first describe the application of this technique to a simple camera motion, and then show how projective duality is used to extend the analysis to a wider class of camera motions and object types that include curved and moving objects.

1 Introduction

One of the fundamental tasks of computer vision is to describe a scene in terms of coherent, three-dimensional objects and their spatial relationships. Computing this description is difficult in part because of the enormous diversity of objects and the almost limitless ways in which they can occur in scenes. A deeper problem is an image's inherent ambiguity: since the process of forming an image captures only two of the three viewing dimensions, an infinity of three-dimensional scenes can give rise to the same two-dimensional image. It follows, therefore, that no single two-dimensional image contains enough information to enable reconstruction of the three-dimensional scene that gave rise to it.

Human vision, on the other hand, routinely circumvents this limitation by utilizing (a) knowledge of scene objects and (b) multiple images, including stereo pairs and image sequences acquired by a moving observer. A typical use of object knowledge is to restrict alternative three-dimensional interpretations of part of an image to a known object or objects. We shall not discuss such techniques here, however. Using more than one image makes it theoretically possible, under certain circumstances, to elicit the three-dimensional scene that gave rise to the images, thereby eliminating the ambiguity inherent in the interpretation of a single image. This power comes at the cost of much more data to process, since there are more images, and the added complexity of additional viewpoints, which if unknown must usually be determined from the images.

We shall be describing a technique for building

*This research was supported by DARPA contracts MDA 903-83-C-0027 and DACA 76-85-C-0004.

a three-dimensional description of a static scene from an extended sequence of images. We started ~ ıesearch with two ideas for simplifying this process. The first was to assume that the camera motion parameters would be supplied to the motion analysis procedure by an independent process, perhaps an inertial-guidance system. Thus, instead of having to estimate both the camera motion parameters and the object locations from the data, as is commonly done in motion-analysis techniques, we decided to concentrate solely on the estimation of object locations. This known-motion assumption is appropriate for autonomous vehicles with inertial-guidance systems and some industrial tasks, such as measuring the height of objects on a conveyor belt passing in front of a camera.

Somewhat surprisingly, this assumption has not been employed as a primary constraint in motion-analysis techniques, and, moreover, has generally been viewed as defining a degenerate and probably uninteresting special case. In stereo analysis, however, this assumption has been applied extensively. The "epipolar" constraint, which reduces the search required to find matching features from two dimensions to one, is derived from the known position of one camera with respect to the other. As far as we know, this epipolar-type constraint has never geen generalized to apply to the three or more images typically required by motion-analysis techniques.

Our second idea was to simplify the matching of features between successive images by taking them very close together. Matching, after all, is one of the most difficult steps in motion processing. This idea, unlike the first, was hardly new. In stereo analysis, for example, it is well known that the difficulty of finding matches increases with the distance between the lens centers. In addition, the accuracy of scene feature estimates improves with the baseline between the cameras. An example of a technique designed to circumvent this trade-off between matching difficulty and expected accuracy is the nine-eyed "slider stereo" procedure developed by Moravec [64]. It achieves a large baseline by tracking features through nine moderately spaced images. We take this idea one step further and collect hundreds of images instead of just a few.

Our willingness to consider hundreds of images was a logical development of our previous research. In a project to recognize objects in range data, we had developed an edge detection and classification technique for analyzing one slice of the data at a time. This approach was adapted to our range sensor, which gathered hundreds of these slices in sequence. The sensor, a standard structured-light sensor, projected a plane of light onto the objects in the scene and then triangulated the three-dimensional coordinates of points along the intersection of the plane and the objects. The edge detection technique located discontinuities in one plane and linked them to similar discontinuities in previous planes.

Although it seems obvious now, it took us a while to appreciate fully the fact that the spacing between light planes makes a significant difference in the complexity of the procedure that links discontinuities from one plane to the next. When the light planes are close together relative to the size of the object features, matching is essentially easy. When the planes are far apart, however, the matching is extremely difficult. This effect is analogous to the Nyquist limit in sampling theory. If a signal is sampled at a frequency that is more than double the signal's own highest frequency, the samples contain sufficient information to reconstruct the original. However, if the signal is sampled less often, the information in the sampled signal is not sufficient for accurate reconstruction. In slices of range data, the objects in the scene have discontinuities that make it impossible to apply the sampling theory directly. However, the basic idea appears sound: there is a sampling frequency below which matching is significantly more difficult.

With our interest in simplifying depth measurement, we decided to take a large number of closely spaced images and see how the matching process was affected. To do this, we borrowed a one-meter-long optical track and gathered 40 images while moving a camera manually along it. For this first sequence, we aimed the camera along the track and moved it straight ahead. Before gathering the data, we had predicted the approximate image speeds for some features in the scene. Afterward, however, it became clear that it would be easier to make such measurements if we aimed the camera perpendicularly to the track instead. We knew that the epipolar lines would then be

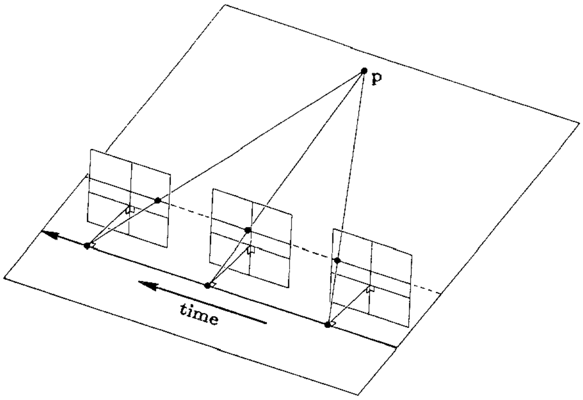


Fig. 1. Lateral motion.

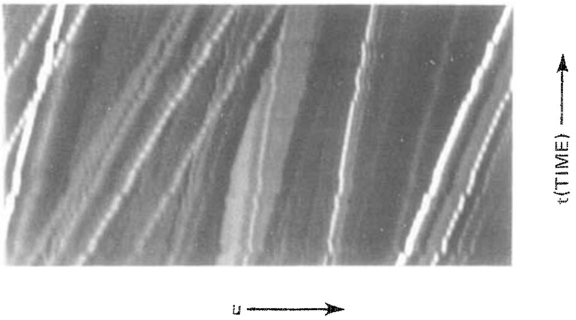


Fig. 3. Spatiotemporal image.

horizontal scan lines, or rows, in the images. Therefore, we gathered a second sequence of images moving right to left along the track, as shown in figure 1. Figure 2 shows the first and last images from this sequence of 32.

We again predicted the image velocities for some of the scene features, one of which was the plant with thin, vertical leaves that appears on the right of the image. To visualize the positional changes caused by the moving camera, we displayed one row from each image. We extracted row 100 from each image and displayed these one above another forming a small rectangular image (see figure 3). This image is a spatiotemporal im-

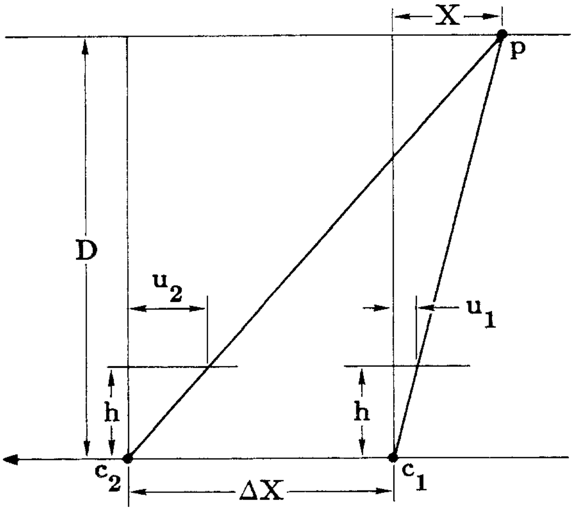


Fig. 4. Lateral motion epipolar geometry.

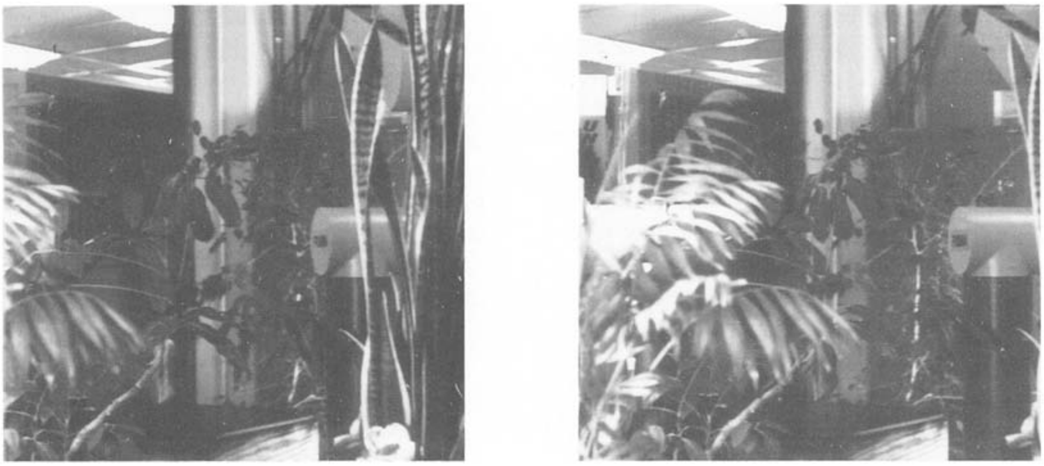


Fig. 2. First and last images from lateral motion sequence.

age in which the spatial dimension is horizontal and the temporal dimension vertical, with time advancing from bottom to top. In constructing this image, we formed what we were to later name an *epipolar-plane image*, or *EPI* for short. (We explain the origin of this name in section 3.) Since then we have learned that other researchers, working independently, have constructed similar images (e.g., Yamamoto [99], Bridwell and Huang [13], Adelson and Bergen [1], and Atherton [4]).

To our surprise, the EPI consisted of simple linear structures that seemed promising prospects for automatic analysis. Even though the spatial images in figure 2, which were used to construct it, contain quite complex shapes and intensity changes, the EPI is essentially composed of homogeneous regions bounded by straight lines. Upon seeing this structure, our first thought was to verify whether the image path of a scene point (i.e., the path of a scene point's projection) formed in this way had to be linear. By analyzing the diagram in figure 4, which illustrates the geometry of lateral motion, we determined that it did. The one-dimensional images are at a distance h in front of the lens centers, while the feature point p is at a distance D from the linear track along which the camera moves right to left. From similar triangles,

$$\begin{aligned}\Delta U = u2 - u1 &= \frac{h^*(\Delta X + X)}{D} - \frac{h^*X}{D} \\ &= \Delta X * \frac{h}{D} \qquad (1)\end{aligned}$$

where ΔX is the distance traveled by the camera along the line, and ΔU the distance the feature moved in the image plane. This expression shows that the change in image position is a linear function of the distance the camera moves. (The jagged appearance of the feature paths in figure 3 is caused by inaccurate manual positioning of the camera on the optical track.)

Equation (1) can be rearranged as follows to yield a simple expression for the distance of a point in terms of the slope of its line in the EPI:

$$D = h^* \frac{\Delta X}{\Delta U} \qquad (2)$$

Given this expression, we outlined the following strategy for building a three-dimensional map of a scene: take a sequence of closely spaced images, form an EPI for each row, locate lines in these EPIs, and compute the three-dimensional coordinates of the corresponding scene features. We elaborate on this process in section 4.

After deriving this depth-from-slope relationship, we considered spatiotemporal paths produced by a camera moving straight ahead rather than laterally. We obtained an equation showing that such paths are hyperbolic. A generalization of this derivation is presented in section 4.

After looking at several EPIs like the one in figure 3, we also observed that EPIs encode the occlusion of one object by another directly. It may therefore be possible to detect higher-level properties of the scene in addition to computing the depths of isolated points. For example, one might mark the occlusion edges of objects, a task that has posed problems for such traditional image-analysis techniques as stereo processing.

In the remainder of this article, we shall explore the concept of epipolar-plane images in more detail. In section 2, we briefly describe related research in the analysis of image sequences. In section 3, we consider the geometric factors involved in the formation of EPIs. In section 4, we describe the results of our experiments analyzing EPIs constructed from lateral motions (diagrammed in figure 1). This discussion includes descriptions of techniques for marking occlusion boundaries, building three-dimensional maps of free space, and transforming imagery to produce EPIs when the camera pans and tilts as it advances in a straight line. In section 5, we show how the principle of projective duality can be used to apply this basic technique to a wider class of tasks, including arbitrary motions in a plane and motions through a world in which the objects may be curved and moving. And finally, in section 6, we birefly discuss the technique's strengths and weaknesses and outline some current and future directions for our work.

2 Related Research

The starting point for most research in image sequence analysis is a model of image formation that predicts the image of a scene feature from the camera's position and orientation. The scene feature is usually a point, but could also be a line, curve, or surface patch, as long as a three-dimensional description of the feature relative to the camera is available. The role of the image formation model is to predict a two-dimensional description of the corresponding image feature: a point, line, curve, or region of smoothly varying intensity.

When the camera moves through a static scene, the image formation model predicts a moving image feature in most cases. Given a suitable collection of moving image features, it is sometimes possible to "invert" the image formation model and to estimate the motion of the camera and the three-dimensional descriptions of the scene features that gave rise to the image features. This process is usually divided into two stages: first, estimation of the image feature's motion; second, estimation of the camera's motion and corresponding scene features.

Our review is divided into four subsections, each of which represents one class of approaches to this problem. The first is based on the interrelationship of differential camera motions, scene features, and the corresponding differential motions of image features. The second samples the camera path at widely spaced locations and attempts to recover changes between camera locations and scene features from the corresponding widely spaced image features. A third approach, which has received far less attention than the first two, considers an image sequence from a densely sampled camera path, and estimates camera motion and scene features from the resulting "paths" of image features; EPI analysis itself belongs to this class. The final approach does not really fit within this framework at all, for it does not require the estimation of image feature motion as a preliminary to estimating camera motion and scene features.

2.1 Techniques Based on Differential Camera Motion

Differential camera motion through a stationary scene induces differential motion of image features. When the image and scene features are points, the differential motion of image features is called the optical velocity field. Research in this area deals for the most part with two issues: estimating the optical velocity field from images, and inferring the scene structure and differential camera motion that induced it.

Estimating the optical velocity field is difficult for a number of reasons. First, local information alone is insufficient, even under ideal circumstances, since only one component of a point's two-component velocity is available directly [35]. Thus, additional constraints are necessary if the problem is to be well posed. Second, since image sequences are sampled in time as well as in space, any spatial or temporal image derivative must be approximated discretely. Finally, images are noisy; differentiating them only aggravates this characteristic, as well as making it more difficult to track points in regions where variations in intensity are of the same order as sensor noise.

A variety of approaches has evolved in an attempt to cope with these difficulties. Horn and Schunk [35] require that optical velocity vary smoothly in regions of the image where intensity is smoothly varying. Nagel [66] describes a more general technique applicable to regions containing edges. Hildreth [31] estimates optical velocity from image contours, rather than from a region of intensities; she too requires that the resulting optical velocity field be smoothly varying. Heeger [30] estimates optical velocity locally by using Gabor filters to reconstruct the spatiotemporal power spectrum of a translating texture.

Interpreting the optical velocity field means estimating not only the differential camera motion, which consists of a translational velocity and an angular velocity, but also, as a rule, aspects of the scene structure. Some work has concentrated on the analysis: determining the relationships among quantities of interest and, wherever possible, finding closed-form solutions. Computational approaches deal more with implementing these solutions and developing strategies to cope with noisy input data. Because estimates of camera motion and scene features are particularly sensitive to image noise when the camera locations are close together, techniques based on differential camera motion pose a special problem.

There has been a great deal of analysis based on local properties of the optical velocity field. Koenderink and Van Doorn [46,47,50] study the relationship between the optical velocity field and its first and second spatial derivatives on the one hand, and geometric properties of the scene (surface orientation, sign of Gaussian curvature, etc.) and the components of camera velocity (translational and angular) on the other. Their emphasis is mainly on qualitative, geometric analysis rather than on algorithms and computations. Longuet-Higgins and Prazdny [55] show how to obtain camera velocity and surface orientation from the optical velocity field and its first and second spatial derivatives. Hoffman [33], who bases his analysis on orthographic instead of perspective projection, uses the optical velocity and acceleration fields, along with the first spatial derivative of the velocity field, to estimate angular velocity and surface orientation for rotation-only camera movement.

Waxman and Ullman [92, 93] derive and implement a method to compute camera velocity as well as the slope and curvature at a point on a scene surface from the optical velocity field and its first and second spatial derivatives at the corresponding image point. Waxman and Wohn [94,95] implement this scheme by estimating the optical velocity field and the necessary derivatives along image contours. Subbarao [78] develops a formalism to estimate camera velocity and surface orientation at a scene point from the first spatial and temporal derivatives of the optical velocity field at the corresponding image point.

All these techniques use only local information in the optical velocity field and thus are sensitive to noise. Other techniques compute motion and structure parameters over larger regions of the field. Prazdny [72] presents a two-stage algorithm that first obtains camera velocity from five image points and their velocities, then computes relative depth and surface orientation from the optical velocity field everywhere in the image. Bruss and Horn [16] employ a least-squares method to estimate camera velocity (but not scene structure) from the entire optical velocity field. Similarly, Prazdny [73] uses the entire optical velocity field to extract the direction of translational velocity. Lawton [51] considers translational motion only and estimates its direction from point features detected by an interest operator; his approach is unusual in that the features are matched and the direction of motion determined simultaneously.

2.2 Techniques Based on Widely Separated Views

Another class of techniques is based on sampling the camera path a widely separated locations rather than differentiating it. Here the problem is usually to match corresponding image features in two or three images and then to infer from them the corresponding scene features, as well as the translation and rotation between the camera locations. Because the camera locations are farther apart than in the case of differential motion, estimates based on corresponding image features tend to be more stable. On the other hand, establishing correspondences between image features is far more difficult, since the widely separated camera locations mean that each feature may have moved to a new position in the image, or the scene feature to which it corresponds may have moved out of the camera's field of view or behind another object in the scene. The larger the range of possible image positions (including not being in the image at all), the more computation is required for each feature.

As in the case of differential camera motion, most research in this area deals with point features in images and scenes; furthermore, it assumes that the correspondences are given. Photogrammetrists were among the earliest to formulate the problem for two or more perspective views of points in space. Thompson [80] develops an iterative solution for two views of five points involving five simultaneous third-order equations. Wong [97] reviews modern photogrammetric techniques for multiple overlapping views of large numbers of points; these techniques also involve iterative solutions of nonlinear equations. Roach and Aggarwal [76] analyze the problem of point correspondence in two and three views and conclude that six correspondences are required to overdetermine the solution if there are two views, and four correspondences if there are three; the equations involved are nonlinear. Ullman [88] deals with point correspondences under orthogonal projection and explores the trade-offs between the minimum number of points and views required for a complete solution.

Longuet-Higgins [53] describes a linear method for solving the problem of two views of eight or more points, although his technique has some stability problems and does not seem to have won over many photogrammetrists. Tsai and Huang [87] independently derive similar results and also consider the question of the uniqueness of solutions. Tsai and Huang [85,86] also consider the problem of two views of a set of points constrained to lie on the same plane; they extract the camera motion by solving a single sixth-order equation and performing a singular value decomposition.

Other authors avoid the use of point correspondences by relying on aggregate features that are more reliable to estimate and easier to put into correspondence. Yen and Huang [101], Liu and Huang [52], and Mitiche, Seida, and Aggarwal [63] all develop solutions for the problem of three views of a set of lines in space. Tsai [83,84] considers two views of a single conic arc. Aloimonos and Rigoutsos [3] combine stereo and motion to estimate the camera motion from an image sequence of coplanar points based on the detection of image points but without the individual correspondences. Kanatani [42,43] uses features computed along entire image contours to estimate and track planar surface motion.

2.3 *Integrating Information Along the Camera Path*

The two approaches discussed so far typically analyze only two or three images at a time. Now we turn to techniques that analyze longer image sequences and, in effect, integrate information from images collected along densely sampled intervals of the camera path. This class of techniques has the potential to combine some of the best characteristics of the approaches already discussed. Since images are sampled densely, image features shift very little from image to image and so correspondences are easier to establish, as in the case of differential camera motion. Since longer image sequences are involved, the camera moves enough to increase the stability of estimates of structure and motion, as in the case of widely separated views. Stability is also improved because more images contribute to these estimates than in either of the other two approaches.

An early example of this approach was provided by Moravec [64], who built a mobile robot equipped with "slider stereo," a camera mounted on a track that takes nine moderately spaced images as it slides across. Webb and Aggarwal [96] process a long sequence of images generated by people moving in a dark room with lights attached to certain of their joints; the goal is to infer jointed motion from the point correspondences. (Hoffman and Flinchbaugh [34] consider a similar problem but use no more than two or three images at a time.) Ullman [90] proposes the "incremental rigidity method," which estimates the three-dimensional structure of a smoothly moving (and possibly smoothly deforming) object incrementally from point correspondences in a sequence of orthographic images. Gennery [26] uses an approach similar to nonlinear Kalman filtering to lock onto and to track a known three-dimensional object. Broida and Chellappa [14, 15] apply a nonlinear Kalman filtering technique to point correspondences in an image sequence to estimate motion and three-dimensional structure.

Buxton and Buxton [17,18] extend the Marr-Hildreth theory of edge detection [59] to spatiotemporal imagery, experiment with detecting edges in closely spaced images of moving blocks, and speculate on the possibility of extracting depth information from the edges tracked over time. Adelson and Bergen [1] emphasize the coherence of image motion in the spatiotemporal block and propose linear spatiotemporal filters to detect such motion. Their figures showing slices of the spatiotemporal block to illustrate simple image motion are strikingly similar to EPIs, although they are not concerned with three-dimensional interpretations.

For Yamamoto [99], the goal is to analyze closely spaced images from a stationary camera viewing a busy street scene. As in EPI analysis, the method involves interpreting each scanline separately over time, although the justification for this decomposition is not clear. The tracking of image points and the detection of occlusions, developed in more detail below, are discussed here as well. Bridwell and Huang [13] report a method of image sequence analysis involving lateral camera motion that constrains the projections of a point in space to a single scanline of the image. However, they used widely spaced images that made it difficult to put image points into correspondence and apparently did not further analyze the image point motion resulting from this camera motion.

2.4 Techniques That Do Not Require Correspondence

A final class of techniques estimates structure and motion from image sequences without first solving the correspondence problem, so that no tracking of image features is necessary. Because the correspondence problem is so difficult if techniques are employed that are based on widely separated camera views—and somewhat less so but still the focus of active research in techniques based on differential camera motion—the possibility of circumventing the problem has great appeal. These "direct methods," as Negahdaripour and Horn [67] and Horn and Weldon [36] call them, are just beginning to be explored, so it is not yet clear how useful they will be.

Blicher and Omohundro [11] compute camera velocity only from first temporal derivatives of intensity at six image locations using Lie algebra methods. Negahdaripour and Horn [67] show how to recover camera velocity relative to a planar surface and the three-dimensional description of the planar surface itself directly from image intensity and its first spatial and temporal derivatives; the solution entails an iterative, least-squares method applied to a set of nine nonlinear equations. Negahdaripour [68] presents a closed-form solution to the same problem that involves a linear system and the eigenvalue decomposition of a symmetric matrix. (Subbarao and Waxman [79] also attack this problem but assume that the optical velocity field is available.) Horn and Weldon [36] propose a least-squares method of estimating camera velocity from image intensity and its first spatial and temporal derivatives; the technique is applicable in cases of pure rotation, pure translation, and when the rotation is known.

2.5 Discussion

The three correspondence-based approaches to motion analysis reviewed above are in essence three ways to interpret the paths of features across images as a camera moves. The first relates the derivative of the feature paths to scene features and to the derivative of the camera path. The second relates samples of the feature paths at wide intervals to scene features and to samples of the camera path at wide intervals. The third relates the entire feature paths to scene features and to the entire camera path.

None of these approaches is without disadvantages. With the derivative approach, establishing the correspondences may be feasible, but stability problems associated with the use of derivatives and small camera motions seem inevitable. With the sampled approach, the correspondence problem is very difficult and possibly even insoluble. The entire path approach avoids these pitfalls at the cost of enormously greater computational requirements. Techniques that do not require correspondence are promising exactly because the correspondence problem is so far from solved, despite all the research devoted to it.

3 Epipolar-plane Images

In this section, we define an epipolar-plane image and explain our interest in it. We first review some stereo terminology and then describe the extension of an important constraint from stereo processing to motion analysis. Next, we use this constraint to define a spatiotemporal image that we call an epipolar-plane image, and show how to construct EPIs from long sequences of closely spaced images. Finally, we briefly restate the approach derived from our original ideas for simplifying depth determination.

3.1 Stereo Terminology

Figure 5 depicts a general stereo configuration.

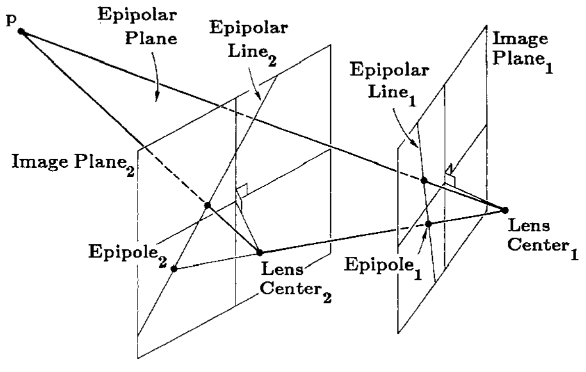


Fig. 5. General stereo configuration.

The two cameras are modeled as pinholes with the image planes in front of the lenses. For each point P in the scene, there is a plane, called the *epipolar plane*, that passes through the point and the line joining the two lens centers. The set of all epipolar planes is the pencil of planes passing through the line joining the lens centers. Each epipolar plane intersects the two image planes along *epipolar lines*. All the points in an epipolar plane are projected onto one epipolar line in the first image and onto the corresponding epipolar line in the second image. The importance of these lines for stereo processing is that they reduce the search required to find matching points from two dimensions to one. Thus, to find a match for a point along an epipolar line in one image, all that is necessary is to search along the corresponding epipolar line in the second image. This geometric relationship is termed the *epipolar constraint*. Finally, an *epipole* is the intersection of an image plane with the line joining the lens centers (see figure 5). When the camera is translating, the resulting epipole is often called the focus of expansion (FOE) because the epipolar lines radiate from it.

3.2 An Epipolar Constraint for Motion

In stereo processing, the epipolar constraint significantly reduces the search required to find matching points. Since we wanted to simplify the matching required for motion analysis, we looked at the possibility of extending this constraint to sequences of three or more images. We found that there is indeed such an extension when the lens center of the camera moves in a straight line. In that case, all the lens centers are collinear, so that all pairs of camera positions produce the same pencil of epipolar planes. A straight-line motion thus defines a partitioning of the scene into a set of planes. The points on each of these planes act as a unit. They are projected onto one line in the first image, another line in the second image, and so on. This partitioning of the scene into planes is a direct extension of the epipolar constraint of two-camera stereo to linear path sequence analysis. To find matches for points on an epipolar line in one image, all that is necessary is to search along the corresponding epipolar line in any other image of the sequence.

The camera can even change its orientation about its lens center as it moves along the line without affecting the partitioning of the scene into epipolar planes. Orientational changes move the epipolar lines around in the images, but, since the line joining the lens centers remains fixed, the epipolar planes remain unaltered.

If the lens center does not move in a line, the epipolar planes passing through a scene point differ from one camera pairing to the next. The points in the scene are grouped one way for the first and second camera positions, a different way for the second and third, and so on. This makes it impossible to partition the scene into a disjoint set of epipolar planes, which in turn means that it is not possible to construct EPIs for such a motion. The arrangement of epipolar lines between images must be transitive for EPIs to be formed.

3.3 Definition of an Epipolar-plane Image

Since the points on an epipolar plane are projected onto one line in each image, all the information about them is contained in that sequence of lines. To concentrate this information in one place, we constructed an image from this sequence of lines. We named this image an *epipolar-plane image* because it contains all the information about the features in one epipolar plane.

Since an EPI contains all the information about the features in a slice of the scene, the analysis of a scene can be partitioned into a set of analyses, one for each EPI. This ability to partition the analysis is a crucial element for our motion-analysis technique. The EPIs can be analyzed independently (possibly also in parallel), and the results then combined into a three-dimensional representation of the scene.

3.4 Construction of EPIs

We began this research with two ideas. The first was to use knowledge of camera motion to reduce the number of parameters to be estimated. However, as a result of our investigation of the geometric constraints that could be derived from

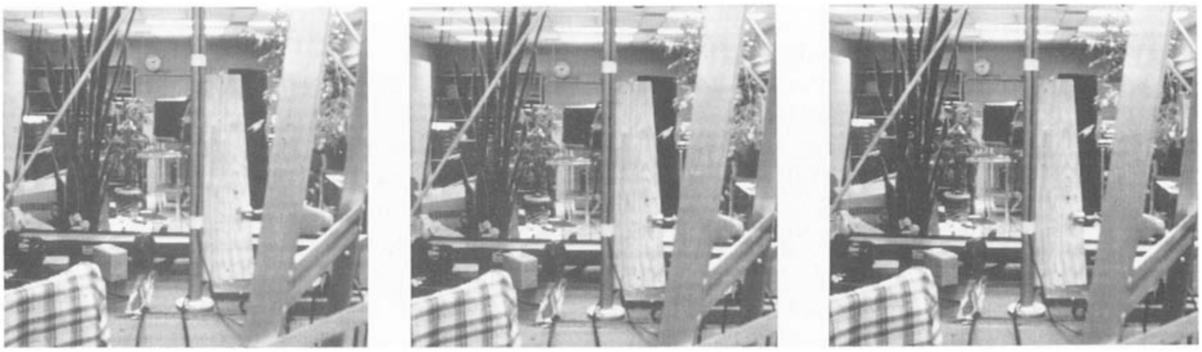


Fig. 6. First three of 125 images.

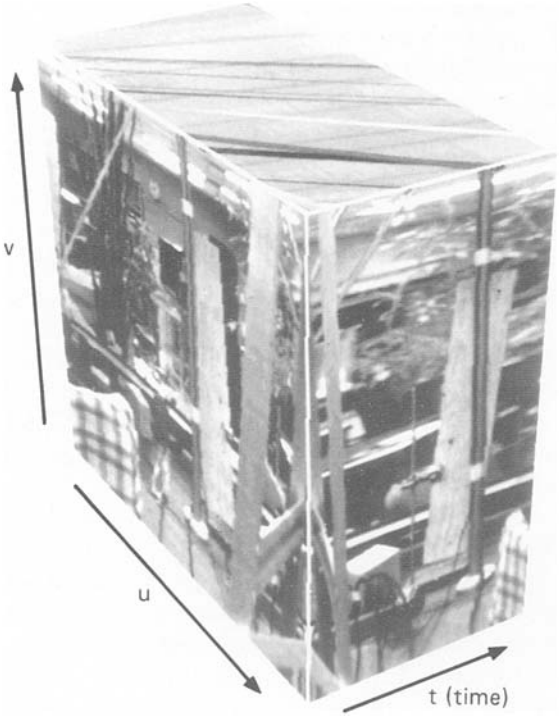


Fig. 7. Spatiotemporal solid of data.

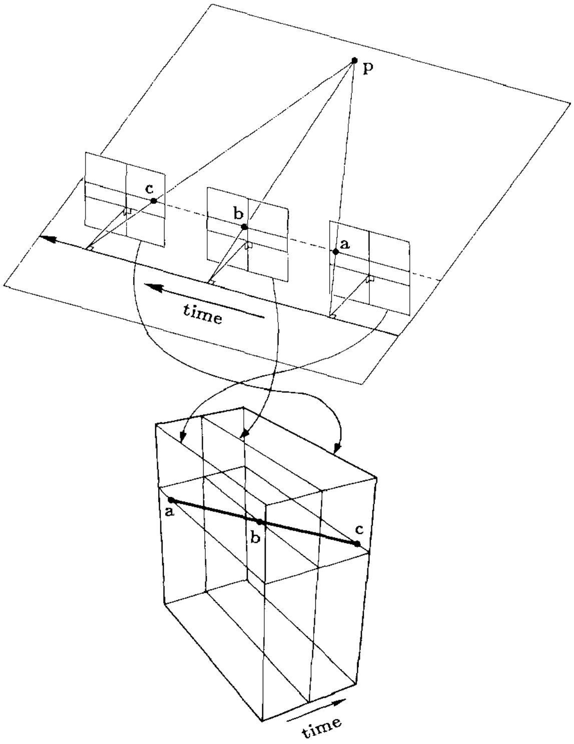


Fig. 8. Lateral motion with solid.

such knowledge, our application of this idea has undergone some revision. Rather than monitor the motion of the sensor, we now restrict the motion to straight lines. This allows us to partition the three-dimensional problem into a set of two-dimensional analyses, one for each epipolar plane.

Our second initial idea was to take long sequences of closely spaced images to obtain a long baseline by tracking features through many similar images. To pursue this idea further than was possible with the few image sequences described in the introduction, we took some longer sequences in which the images were so close together that no single image feature moved by more than a few pixels from image to image. (Figure 6 shows the first three images from one of our sequences of 125.) This sampling frequency guaranteed a continuity in the temporal domain similar to that of the spatial domain. Thus, an edge of an object

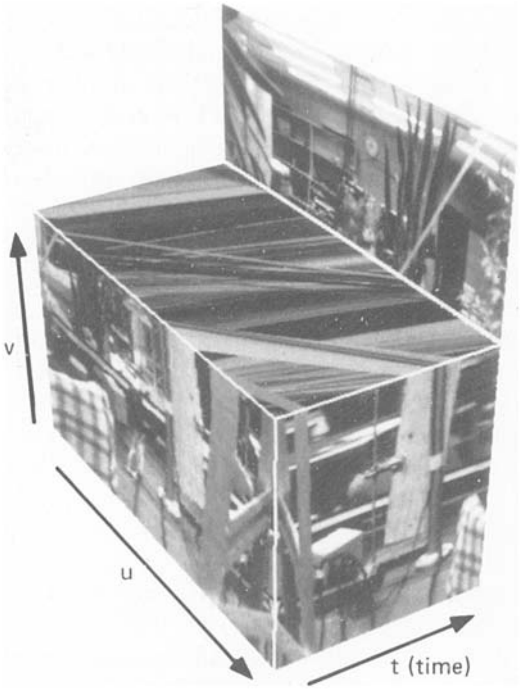


Fig. 9. Sliced solid of data.

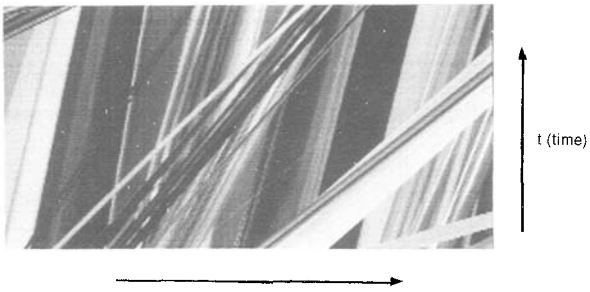


Fig. 10. Frontal view of the EPI.

in one image appeared temporally adjacent to (within a pixel of) its occurrence in both the preceding and following images. This temporal continuity made it possible to construct a solid block of data in which time is the third dimension and continuity is maintained over all three dimensions (see figure 7). This solid of data is referred to as *spatiotemporal data*.

An EPI is a slice of this solid of data. The position and shape of the slice depend on the type of motion. An EPI for a *lateral motion*, whereby the camera is aimed perpendicularly to its path, is a horizontal slice of the spatiotemporal data. This is because the epipolar lines are horizontal scanlines that occur at the same vertical position in all images (see figure 8). Figure 9 shows one of these slices through the solid of data in figure 7. Figure 10 is a frontal view of that slice. In this image, time progresses from bottom to top and, as the camera moves from right to left, the features shift toward the right.

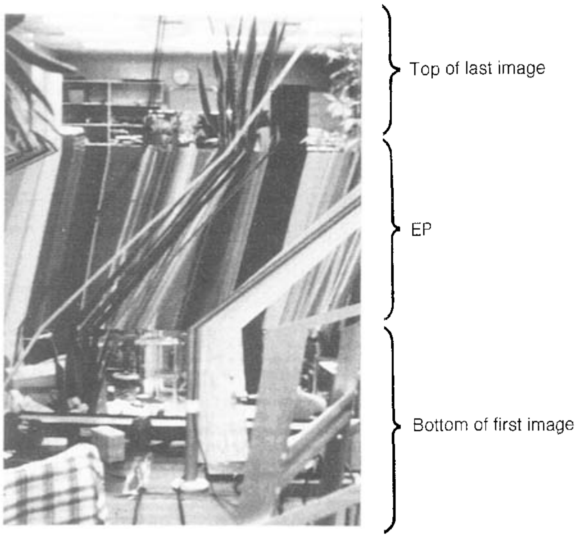


Fig. 11. EPI with portions of the spatial images.

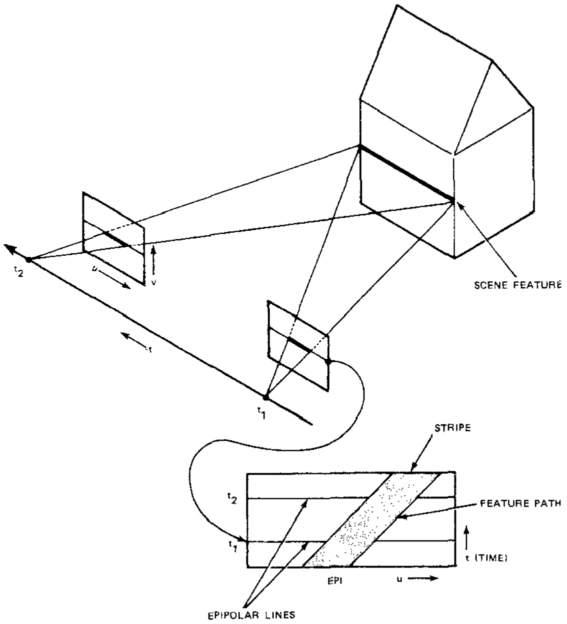


Fig. 12. Relationship of scene features to feature paths.

Figure 11 is another way of displaying the EPI in figure 10 that makes it easier to identify objects in the scene that are associated with patterns in the EPI. In this figure, the lower part of the first spatial image is displayed below the EPI, while the upper part of the last spatial image appears above it. Note, for example, that the tall, thin-leafed plant, which is visible at the top of the picture, is associated with the dark parallel lines that start in the lower left corner of the EPI. The closest leg of the ladder produces the white stripe at the lower right of the EPI.

To be more precise about what we call a feature in an EPI, consider figure 12, which shows the relationship between the patterns in an EPI and the objects intersected by the corresponding epipolar plane. The front wall of the house in the scene produces a stripe in the EPI. We refer to the edges of this stripe as *feature paths* or *paths* and the corresponding points on the sides of the block as *scene features,* or *features.* Our motion-analysis technique locates feature paths in an EPI and then uses properties of these paths to estimate the three-dimensional coordinates of the corresponding scene features.

For a feature path to be continuous in an EPI, the images in the sequence have to be taken closely enough together to make the stripe in figure 12 continuous. If the images are too far apart, the stripe degenerates into a sequence of disconnected sections (see figure 13). Therefore, three factors affect the continuity of feature paths in an EPI: the width of the object in the scene, the distance of the object from the camera path, and the distance between lens centers. To guarantee the continuity of paths associated with thin objects, we generally take images closely enough together so that nothing moves more than a few pixels between images.

It is easy to construct EPIs for motions in which the camera is aimed perpendicularly to its trajectory because the epipolar lines are image scanlines. However, if the camera aims forward or changes its orientation as it moves, the construction is more complicated. Figure 14 illustrates the sequences of epipolar lines produced by three different types of motion. The top sequence shows the simple case of perpendicular viewing. The middle one shows the case in which the camera is aimed at a fixed angle relative to its path. The third sequence illustrates the case in which the camera pans, tilts, and rolls as it moves. The images to the right are EPIs from these cases.

In the first case, an EPI is constructed by extracting one row of each image and inserting it into a new image. For the second case, the construction is a little more difficult because the epipolar lines are not horizontal. The epipole, however, is at a fixed position in the images, since the camera is at a fixed orientation relative to its trajectory. (Recall that the epipole is the intersection of the line between the lens centers and the images.) In this case, the epipolar lines associated with an epipolar plane are at a fixed angle and radiate from the epipole. To construct an EPI, one must extract a line at a specific angle from each image and insert it into a new image. Since the lines form the same pattern in all images, their extraction can be quite simple. For this type of motion, an EPI is a planar slice through the solid of data that passes through the epipoles and is at different angles for different epipolar planes.

For the most complicated case, i.e., when the camera is altering its orientation as it moves, the position of the epipole changes from one image to the next. In addition, the orientations of the corresponding epipolar lines change, and this makes their extraction more difficult. In this case, an EPI corresponds to a slice through the solid of data that undulates and shifts as the camera's orientation changes. In sum, we can say that it is possible to construct EPIs for all straight-line motions, but there are three levels of difficulty in doing so.

3.5 *Discussion*

Both of the ideas we started will lead to ways of simplifying the matching of features, the most difficult task in motion analysis. The first idea—

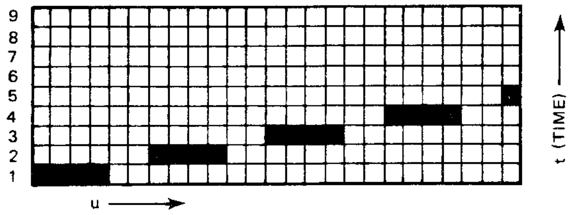


Fig. 13. Discontinuous "stripe" in an EPI.

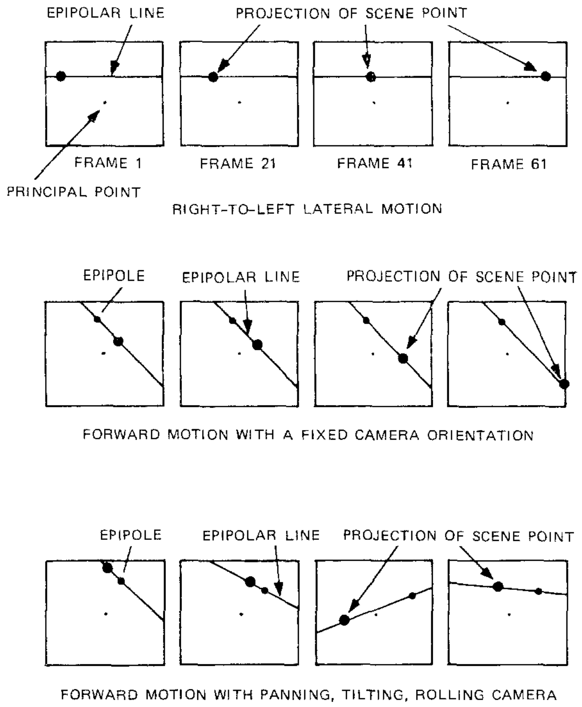
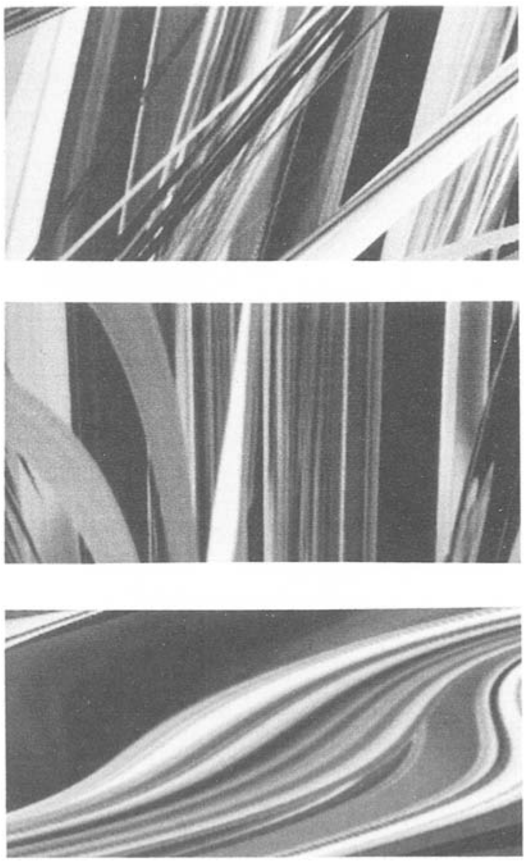

EPIPOLAR LINE
PROJECTION OF SCENE POINT
FRAME 1
FRAME 21
FRAME 41
FRAME 61
PRINCIPAL POINT
RIGHT-TO-LEFT LATERAL MOTION
EPIPOLE
EPIPOLAR LINE
PROJECTION OF SCENE POINT
FORWARD MOTION WITH A FIXED CAMERA ORIENTATION
EPIPOLE
EPIPOLAR LINE
PROJECTION OF SCENE POINT
FORWARD MOTION WITH PANNING, TILTING, ROLLING CAMERA

Fig. 14. Sequences of epipolar lines for different types of motion.



using knowledge of the camera's motion to reduce the number of unknown parameters—led us to restrict that motion to straight lines. This made it possible to partition the three-dimensional problem into a set of two-dimensional problems. Combining this tactic with the second idea of working with a sequence of closely spaced images changed the problem from one of matching features in spatial images to finding feature paths in spatiotemporal images. This approach led to a technique that employs spatiotemporal processing to determine the locations of object features in disjoint two-dimensional slices of the scene, and then combines these separate results to form a three-dimensional description of the scene.

4 Linear Camera Motion

In section 3, we explained how the analysis of straight-line motion sequences could be decomposed into a set of planar analyses. In this section, we describe the techniques of planar analysis. We start by obtaining an expression that describes the shape of a feature path in an EPI derived from a linear motion in which the camera is aimed at a fixed angle relative to its path. Then, after discussing occlusions and free space, we describe the results of an experimental system for analyzing EPIs produced by lateral motion. Finally, we describe the analysis for a camera looking at some arbitrary angle relative to its path.

4.1 Feature Path Shapes

In an epipolar plane, scene points are two-dimensional and the image is formed along a line, so we refer to the image *line* instead of the usual image plane. For the time being, let the camera

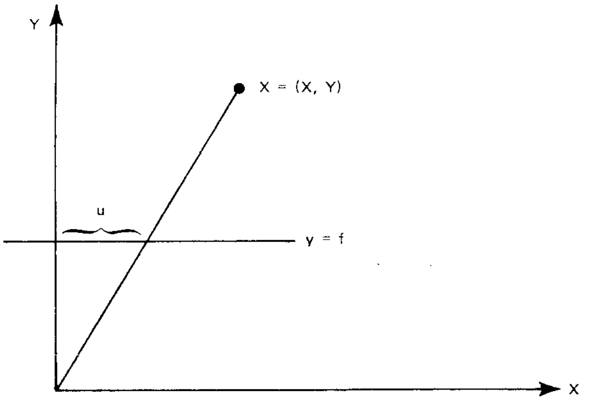


Fig. 15. One-dimensional camera geometry.

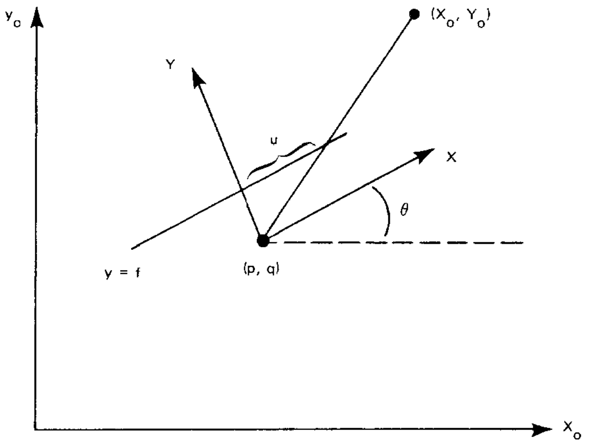


Fig. 16. Rotated and shifted one-dimensional camera.

center (which, in this context, is called the center of projection) be at the origin, a scene point be at $\mathbf{x} = (x, y)$, and the image line be $y = f$ (see figure 15). The central projection of $\mathbf{x}$ onto the image line is simply the intersection of the ray from the origin through $\mathbf{x}$, which is $(fx/y, f)$. We drop the second coordinate, since it is the same for all image points, and take the coordinate along the image line to be $u = fx/y$.

If the scene point's coordinate system differs from the camera's, we must first transform the scene point into the camera's coordinate system and then project as before. This is the situation depicted in figure 16. As before, the internal camera coordinate system is defined with the camera center at the origin and the viewing direction along the positive y-axis. Let the camera center be at $\mathbf{p} = (p, q)$ in the global coordinate system and let the camera coordinate system be rotated counterclockwise by θ. The relationship between a point in the global coordinate system $\mathbf{x}_0$ and one in the camera coordinate system $\mathbf{x}$ is

$$\mathbf{x}_0 = R\mathbf{x} + p \tag{3}$$

so that

$$\mathbf{x} = R^t(\mathbf{x}_0 - \mathbf{p}) \tag{4}$$

where the rotation matrix R is

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{5}$$

As before, the projection of the point $\mathbf{x}$ is $u = fx/y$; using equation (4) we find that, in terms of the global coordinates of the scene point,

$$u = f\left(\frac{(x_0 - p)\cos\theta + (y_0 - q)\sin\theta}{(p - x_0)\sin\theta + (y_0 - q)\cos\theta}\right) \tag{6}$$

If the camera is moving and is aimed in a fixed direction, its position $\mathbf{p}$ is a function of time; because equation (6) is still valid, u will be also a function of time.

Without loss of generality, we take the camera path to begin at the origin, at time $t = 0$, moving along the positive x-axis (see figure 17). With a

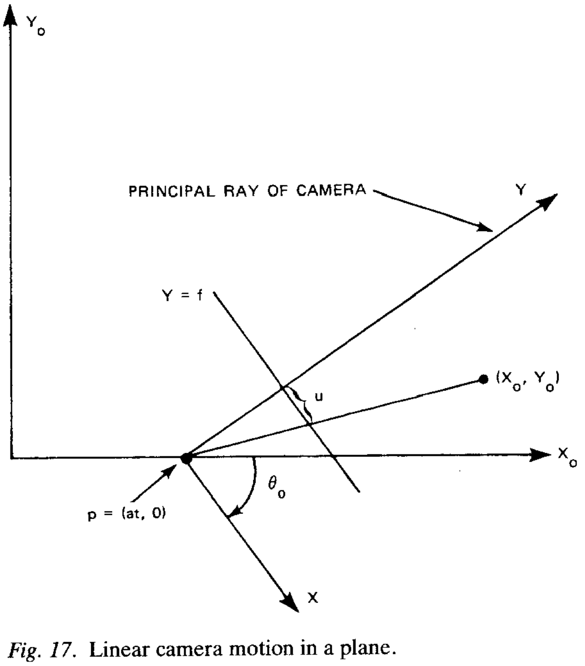


Fig. 17. Linear camera motion in a plane.

slight loss of generality, we let this movement be at a constant speed a, so that $\mathbf{p} = (at, 0)$. If we substitute this choice of $\mathbf{p}$ into equation (6), we find, after some rearranging, that

$$aut\sin\theta_0 + (y_0\cos\theta_0 - x_0\sin\theta_0)u + aft\cos\theta_0 - f(x_0\cos\theta_0 + y_0\sin\theta_0) = 0 \quad (7)$$

Note that a perpendicular viewing direction means that $\theta_0 = 0$, so that equation 7 reduces to

$$y_0u + aft - x_0f = 0 \quad (8)$$

The feature paths are therefore linear.

The linearity of feature paths in epipolar images greatly simplifies their analysis. The algorithm for each epipolar image's initial processing is as follows: detect and link edges, fit straight-line segments to the edges, and then use the parameters of each line segment to estimate the position of the corresponding scene point. This estimate is obtained quite simply from equation (8). Consider a typical epipolar image, with the image coordinate u on the horizontal axis and t on the vertical. If we then rearrange equation (8) in standard slope-intercept form, as in

$$t = \left(\frac{-y_0}{af}\right)u + \frac{x_0}{a} \quad (9)$$

the slope of the line gives the corresponding scene point's perpendicular distance from the camera path, while its u-intercept gives the distance along the path.

Note that no matter what the value of a, the speed of the camera, may be, a line that is more vertical than another corresponds to a scene point whose perpendicular distance from the camera path is greater than that of the scene point corresponding to the less vertical line. This is because the larger the magnitude of the slope, the more vertical is the line. This agrees with our intuition that image points that traverse the image more rapidly correspond to closer scene points. Also, it should be noted that the direction of motion, or sign of a, determines the sign of the slopes of the image paths. If the camera moves along the positive x-axis and looks to its left ($\theta = 0$), we expect points in the image to move from right to left, which means the lines have a negative slope. In this case, a is positive and, since a real camera can see only what is in front of it, y_0 is always positive, so obviously a line given by equation (9) has a negative slope.

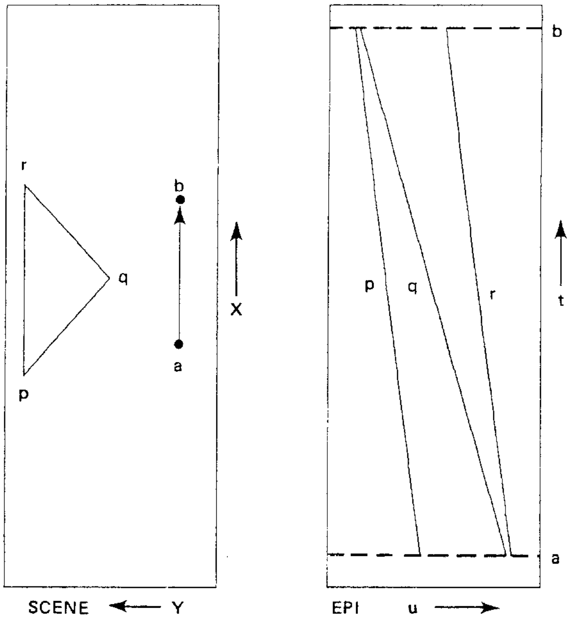


Fig. 18. A simulated planar scene.

Consider the simulated planar scene in figure 18. The scene and camera path are on the left, and the feature paths are on the right. The scene consists of the triangle **pqr**. The camera path is a straight line from **a** to **b**, with the camera's viewing direction perpendicular to the path and toward the triangle. In the EPI on the right, the spatial dimension, which is the u-coordinate, is on the horizontal axis and the temporal dimension is vertical, increasing upward. Both image formation and edge detection are simulated; in effect, the camera "sees" only the vertices of the triangle. The feature paths in the EPI are linear, as expected. The one corresponding to **q**, which is the point closest to the camera path, is the least vertical. The other two paths, which correspond to scene points at the same depth from the camera path, are parallel but have different u-intercepts, reflecting the difference in their distances along the camera path.

Note that in the simulated EPIs, such as the one in figure 18, the feature paths have negative slopes, unlike the paths in the real EPI in figure 10. This difference lies in the camera's viewing direction relative to its path. In the simulation the camera was aimed to the left, causing features to

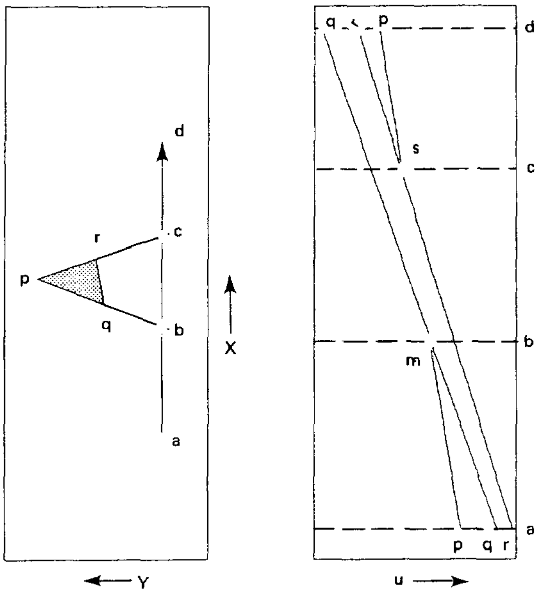


Fig. 19. Occlusions and disocclusions.

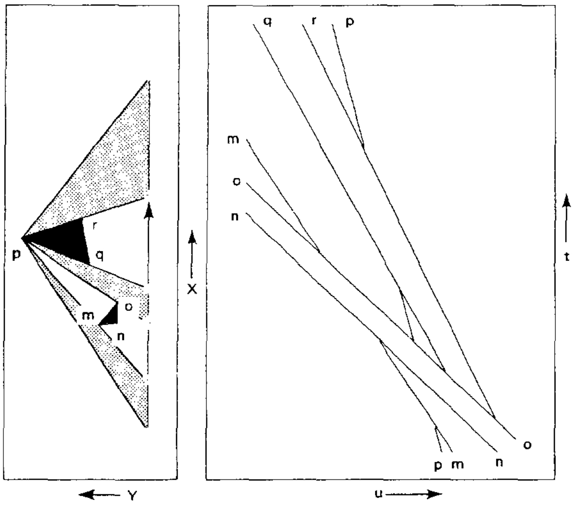


Fig. 20. Scene feature visible three times.

move from right to left across the camera's image plane, while in the case of the real EPI the camera faced toward the right, so the feature paths ran from left to right as the camera progressed.

4.2 Occlusions and Disocclusions

Occlusions and disocclusions, the emergence into visibility of a point from behind an occlusion, induce a branching structure in the image feature paths, as depicted in figure 19. On the left is a planar scene containing a triangle and a straight path along which the camera moves from **a** to **d**, again looking perpendicular to the path toward the triangle. On the right are the image point paths. Two types of branches exist. Both are Y's formed from *two* line segments. They differ, however, in their orientation relative to the direction of motion. The first is a *merge*, like the point **m** in the image, where one line segment meets another and stops, while the other continues. The other is a *split*, like the point **s** in the image, where a line segment is spawned by an existing one, which continues.

A merge corresponds to an occlusion of one scene point by another, while a split is the emergence into visibility of a previously occluded scene point. Consider the situation in figure 19. Between points **a** and **b**, the camera can see all three vertices of the triangle. Between **b** and **c**, the point **p** is occluded; between **c** and **d**, all three vertices are again visible. In the image on the right, all three vertices are visible until **m**, at which juncture the image feature path corresponding to **p** merges into that corresponding to **q**. This occurs when the camera is at **b**, which is collinear with **p** and **q**, so that their images must be identical. Past **b**, **p** is occluded, so its image feature path terminates; **q** is still visible, however, so it continues. Only two image point paths are present until **s**, when the image feature path corresponding to **p** splits off from that corresponding to **r**. This occurs when the camera is at **c**, which is collinear with **p** and **r**, so their images are identical. Past **c**, **p** is visible, so its image feature path begins again, and all three image feature paths are present until the end of the camera path.

Since an occlusion of one point by another always involves occlusion by the closer point of the one farther away, the upside-down "Y" corresponding to the occlusion always consists of the more horizontal path's cutting off the more vertical one. Conversely, the disocclusion of a point formerly occluded by another always induces a rightside-up "Y" where the more vertical path splits off from the more horizontal one.

One advantage of analyzing long sequences of images is that a scene feature may be visible several different times along a path. Figure 20 illustrates a situation in which point **p** is visible three times, as indicated by the shaded regions. If the processing can identify the three line segments in the EPI as belonging to a single feature, it can compute a significantly more precise estimate of the feature's location than if it used but a single segment.

4.3 The Free Space Map

It may seem at first glance that the topology of the epipolar-plane image, as determined by the splits and merges, is irrelevant to the goal of building a map of the scene. After all, estimating the parameters of a linear image feature path enables us to estimate the location of the corresponding scene point, so what is to be gained by devoting attention to the question of which points were visible at any given moment?

The problem is that we can estimate the parameters of linear image feature paths with reasonable accuracy only at image edges, which occur relatively sparsely in the image. To rely completely on these parameters means that our map of the scene will be sparse as well. The advantage of using these split–merge events and their implications is that they provide information about what is happening in the scene at points other than those that result in image edges.

The essential idea here is that, when a scene point is visible, the line of sight from the camera center to the point must intersect no other objects (we exclude the possibility of nonopaque objects). If images are acquired continuously as the camera moves around a scene, the line of sight to a given scene point sweeps out the three-sided area formed by the path and the initial and final lines of sight. If nothing is moving but the camera, and the feature is viewed continuously, this area then contains no objects; in other words, it is free space.

Others have used similar ideas to construct a map of free space. Some who have taken more sophisticated approaches involving the use of probabilistic knowledge are Moravec and Elfes [65] (from sonar rather than intensity images) and Chatila and Laumond [19]. Bridwell and Huang [13], mentioned earlier in connection with their use of straight camera paths and perpendicular viewing directions, also computed a map of free space.

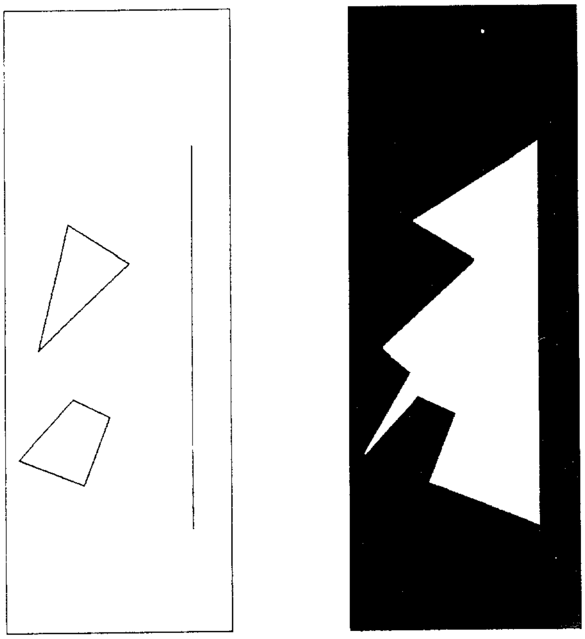


Fig. 21. The free-space map for a simple scene.

Here we take a very simple approach to the construction of a map of the free space in a scene. The area swept by lines of sight must be approximated, since images cannot be acquired and processed continuously; instead the images are acquired at a sampling of positions along the camera path. If this sampling occurs frequently relative to the spacing of objects in the scene, it is reasonable to approximate the free space by computing it as if points visible in consecutive images had been continuously visible along that portion of the camera's path between which the two images were formed. Figure 21 illustrates this process for a simple scene. The fact that images must be taken in rapid succession to obtain an accurate estimate of free space is another argument for the approach employed here and, at the same time, against the practice of relying exclusively on widely spaced images.

Note that the computation of free space depends on knowing the camera path and the location of the scene points. The free space itself is the union of all areas in the scene through which the camera has observed features, so the complement of free space is the area in the scene through which no features have been seen. The complement of free space has as part of its boundary a subset of those scene points whose locations are known, but little information is available on the other points in this "not free space." For example, the "not free space" could contain any opaque object that does not result in image edges (for example a rear wall), and the imagery would be identical.

4.4 Experimental Results

We have implemented a sequence of programs to explore the techniques described in the previous sections. Here, we briefly describe two versions of a program to build three-dimensional descriptions of scenes by analyzing EPIs constructed from lateral motion. The first version of the program consisted of the following steps:

1. Three-dimensional convolution of the spatiotemporal data.
2. Slicing the convolved data into EPIs.
3. Detecting edges, peaks, and troughs.
4. Segmenting edges into linear features.
5. Merging collinear features.
6. Computing x–y–z coordinates.
7. Building a map of free space.
8. Linking x–y–z features between EPIs.

In this section, we illustrate the behavior of this program by applying it to the data shown in figure 6.

The first step processes the three-dimensional data to determine the spatiotemporal contours to be used subsequently as features (as well as, incidentally, to reduce the effects of noise and camera jitter). This is done by applying a sequence of three one-dimensional Gaussians (see also Buxton and Buxton [17]).

The second step forms EPIs from the convolved spatiotemporal data. For a lateral motion this is straightforward, as the EPIs are horizontal slices of the data. Figure 22 shows an EPI selected to illustrate steps 3–7. This slice contains a plant on the left, a shirt draped over the back of a chair, part of a tabletop, and, in the right foreground, a ladder.

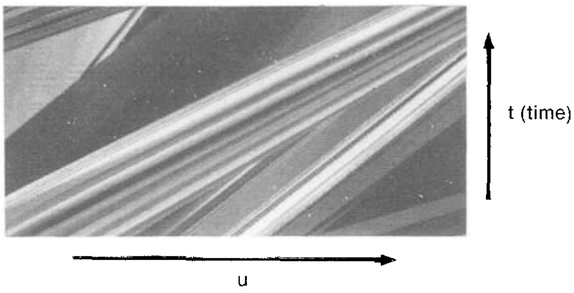


Fig. 22. EPI to be analyzed.

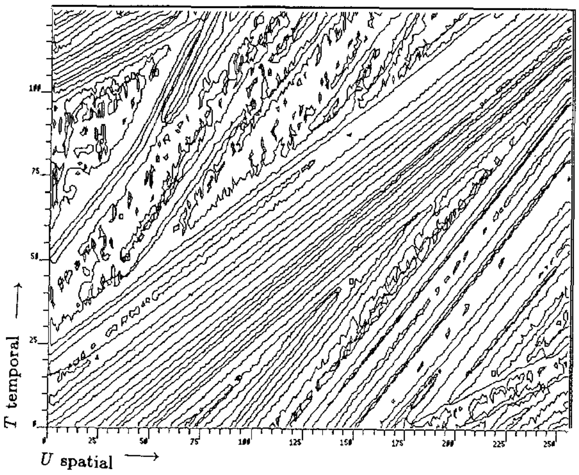


Fig. 23. Edge features in EPI.

The third step detects edgelike features in the EPI. It locates four types of features in the difference of Gaussians: positive and negative zero crossings (see Marr and Hildreth [59]) and peaks and troughs. A zero crossing indicates a place in the EPI where there is a sharp change in image intensity, typically at surface boundaries or surface markings; a peak or trough occurs between a positive and a negative zero crossing. Zero crossings are generally more precisely positioned than peaks or troughs. Figure 23 shows all four types of features detected in the EPI shown in figure 22.

The fourth step fits linear segments to the edges. It does this in two passes. The first pass partitions the edges at sharp corners by analyzing curvature estimates along the contour. The second pass applies Ramer's algorithm (see Ramer [75]) to recursively partition the smooth segments into linear ones. Color figure 56 shows the linear segments derived from the edges in figure 23. The segments are color-coded according to the feature type that gave rise to them: red, negative zero

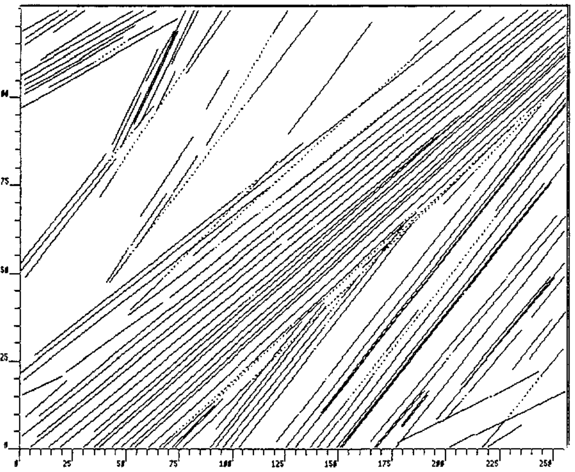


Fig. 24. Merged lines.

crossing; green, positive zero crossing; blue, peak; and yellow, trough.

The fifth step builds a description of the line segments that links together those that are collinear. The purpose is to identify sets of lines that belong to the same feature in the scene. By bridging gaps caused by occlusion, the program can improve its estimates of the features' locations as well as extract clues about the nature of the surfaces in the scene. The dashed lines in figure 24 show those linear features that are linked together. Line intersections indicate temporal occlusions. For each intersection, the feature with the smaller slope is the one that occludes the other.

The sixth step computes the x–y–z locations of the scene features corresponding to the EPI features. The scene coordinates are determined uniquely by the location of the epipolar plane associated with the EPI as well as with the slope and intercept of the line in the EPI. To display these three-dimensional locations, the program plots the two-dimensional coordinates of the features in that particular epipolar plane. Figure 25 (left) shows the epipolar plane coordinates for the features shown in figure 56. In figure 25 (right), the ellipses indicate estimates of the error associated with each scene feature. The horizontal line across the bottom of the figures indicates the camera path.

The scene feature errors are estimated as follows. Each scene feature is related in a simple way to the parameters of the corresponding line in an EPI. However, errors in detecting the image features to which the line is fitted result in errors in the line parameters and thus an error in the scene feature. To estimate the scene feature errors, we assume that the error in detecting image features is Gaussian and approximate the relationship between scene and image features with a linear function. It is therefore natural to approximate the error in the scene feature estimate with a two-dimensional Gaussian. (For details on this simple application of estimation

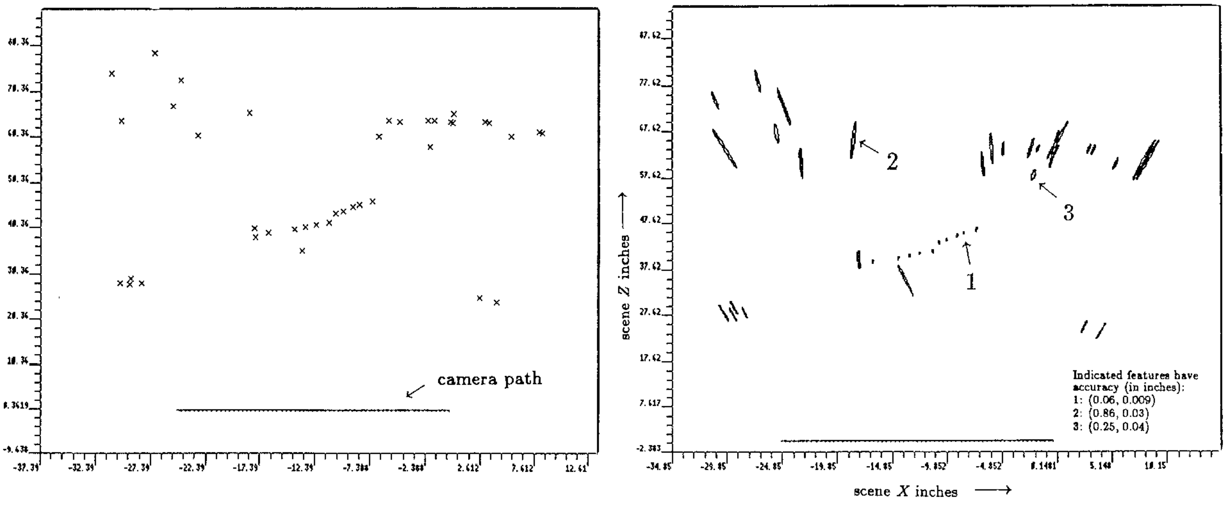


Fig. 25. x–z locations (*left*); 99% confidence ellipses (*right*).

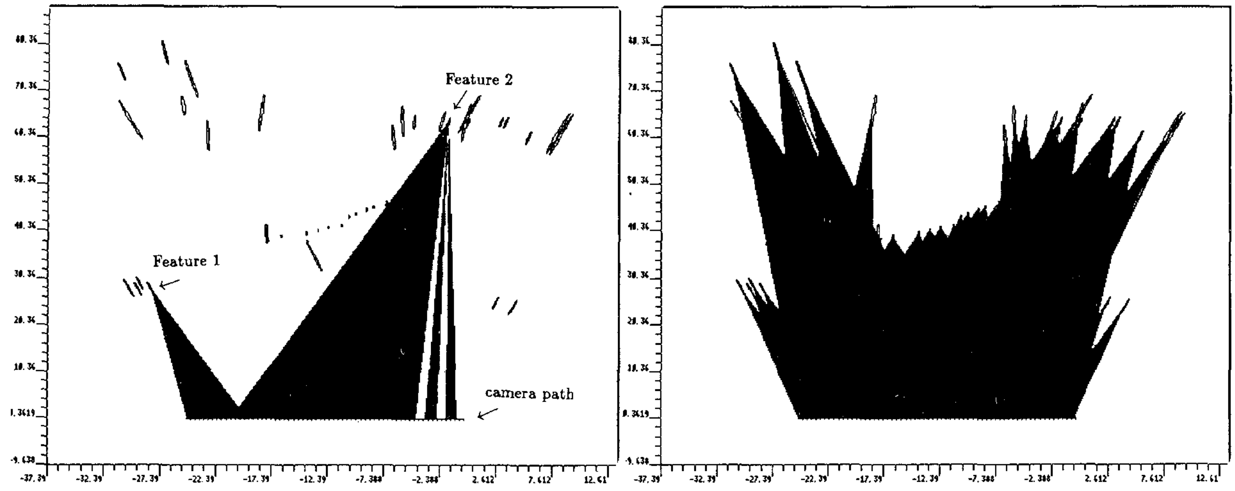


Fig. 26. Free space.

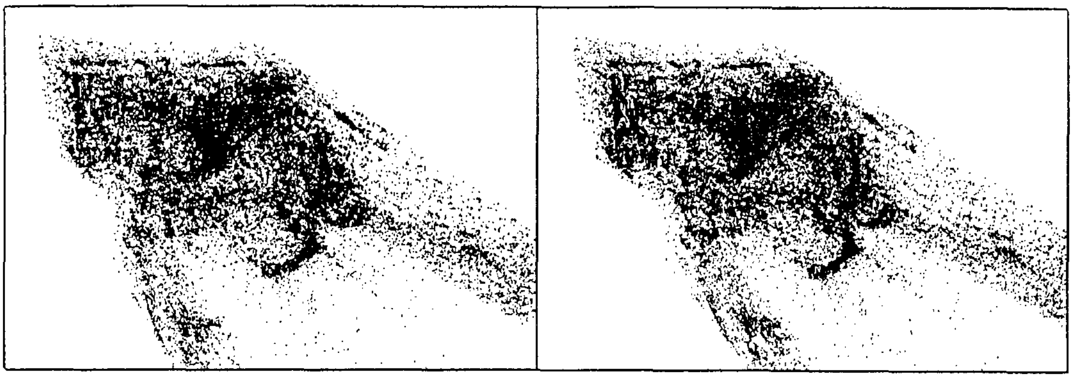


Fig. 27. Crossed-eye stereo display of all x–y–z points.

theory, see Beck and Arnold [9] or Melsa and Cohn [62].) Each ellipse in figure 25 (right) is the contour of equal likelihood at the 3-standard-deviation level, so that, if all of our assumptions are correct, the probability that the true scene feature lies inside the ellipse is greater than .99. In general, more observations of a scene feature reduce the size of its error ellipse, and observations over a longer baseline reduce its eccentricity.

The seventh step builds a set of two-dimensional maps of the scene, indicating regions that are empty. This construction is demonstrated for a few points in figure 26 (left). As mentioned, the principle here is that, if a feature is seen continuously over some interval by a moving camera, then nothing is occluding it during that motion. Since nothing occludes it, nothing lies in front of it, and the triangle in the scene defined by the feature and its first and last points of observation constitutes empty space. We build a map of this free space by constructing one of these triangular regions for each line segment found in an EPI, and then or'ing them together. In figure 26 (left), the black triangles are free space; they are all bounded on one side by the camera path, with the opposing vertex of each at the feature. Note that the left feature is viewed just once, while the other is seen in three distinct intervals, thus yielding three free space triangles. The row of small ellipses across the middle is the shirt visible in the foreground of figure 6. Figure 26 (right) shows

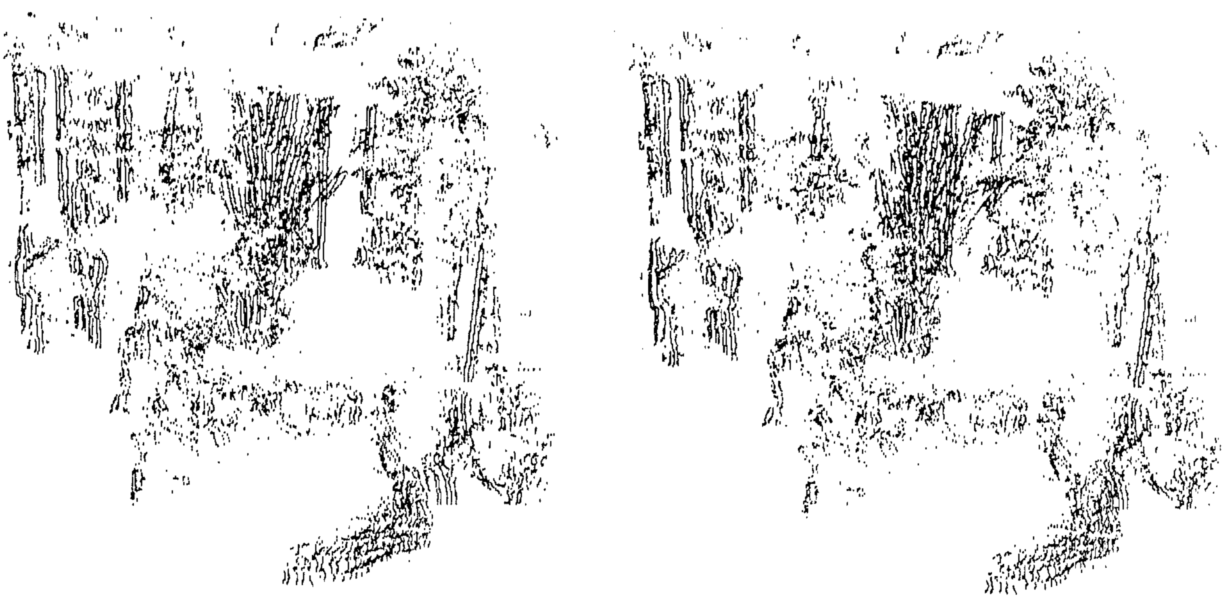


Fig. 28. Crossed-eye stereo display of filtered x–y–z points.

the full free space map constructed for the EPI features of figure 25. An estimate of free space volume could be obtained by combining the free space regions of individual EPIs over some vertical interval. Such a free space volume would be useful for navigation; a vehicle could move freely in that volume, knowing that it would run not into obstacles.

Figure 27 is a stereo (crossed-eye) display, showing the full set of points derived from all the EPIs. The display is relatively dense, since all points, including those arising from quite short segments, are depicted. The eighth step filters these matches by eliminating those that do not have similar features in adjacent EPIs. It links features between adjacent EPIs if their error ellipses overlap, and discards features that do not have at least one matching feature from an adjoining EPI. Figure 28 displays the filtered set of features. Figure 29 shows, for reference, the actual scene from nearly this perspective. Note the shirt, the chair at the middle left, the tall plant in the center, and the diagonal bar. Color figure 57 shows a color-coded depiction of the scene features visible from frame 123.[1]

[1] It will be obvious in looking at some of the figures in this article that we have come up against the standard graphics display problems in this work. In research of this sort, where massive amounts of data are being manipulated, it is crucial to be able to display results graphically. The intrinsic three-dimensionality of our results makes this especially difficult. In our laboratory, we have tools that make possible rapid display of image sequences (using motion to induce the perception of depth), both anaglyphic and polarizing displays of stereo pairs (for perception of stereoscopic depth), and ample use of color for coding other dimensions. Many of these techniques are not appropriate for print and, even in the laboratory, people vary in their preferences among different display methods. Our display techniques include the following. In the laboratory, we can view the original data as a sequence, presenting successive frames at near-video rates. For viewing estimates of scene points, we can display color-coded superpositions of points over the imagery, and view these as a dynamic display. Color figure 57 is one frame from such a display. The colors indicate distances of scene points from the camera: green, less than than 76 inches; red, between 77 and 153; yellow, between 154 and 229; purple, between 230 and 306; and red, between 307 and 382. A similar display can be made for the occlusion edges, as shown in color figure 59. Similarly, we can build rotating displays of isolated or linked 3-D points, one frame of which can be seen as figure 29. None of these adapt well to either print or NTSC videotaping: in videotaping, resolution is an issue, and our displays tend to be quite large. When we are interested in displaying an isolated part of the results, we find the form of color figure 58 to be our most successful. That figure shows a voxel display of the nearest tree of the outdoor sequence, and is again one frame of a dynamic display. We expect to have an even bigger job in showing our processing stages as we continue the development of our current spatio-temporal surface version of the EPI analysis.

Fig. 29. Scene viewed from perspective similar to above.

The second version of this program differs from the first in four ways. First, it no longer detects and analyzes peaks and troughs. These do not necessarily correspond to scene features. They are as likely to be artifacts of the zero crossing process as actual details, and, if they are the latter, are extremely sensitive positionally to variations in illumination.

The second difference is that the new program does not link features between EPIs. We found that, to track vertical connectivity reliably, a model of the expected spatial variation of features is necessary in addition to the error ellipses. For example, features on the shirt of figure 28 are positioned to an accuracy, in x and z, of a few hundredths of an inch, so their error ellipses are quite small. At this distance, adjacent epipolar planes have a vertical separation of about a quarter of an inch, while the pattern on the shirt has a lateral drift of up to about a tenth of an inch. This means that features may shift laterally several times their accuracy limits between EPIs. Instead of continuing with this approach of trying to recompute spatial connectivity, we are exploring a technique for detecting the connectivity directly from the spatiotemporal data. In this work we locate spatiotemporal *surfaces* (3-D), rather than *contours* (2-D), in the three-dimensional solid of data. A surface description has connectivity both temporally (as in the EPI analysis above) and spatially (as is apparent in a normal image); the temporal components enable the EPI analysis, while the spatial components allow vertical integration (plus other advantages, such as increased feature-tracking support, a capability for handling features lying horizontally in the EPI, and a structure that we think will be crucial for moving to nonlinear camera paths). As this three-dimensional analysis has not yet been completed, it is not included in this second version of the program.

The third difference is the incorporation of a process to mark the occlusion boundaries of scene objects. This is done by analyzing the patterns of line intersections in the EPIs. Since most of the intersections are only implicit in the set of lines produced by the earlier analysis, this step extends each line until it intersects another (see figure 30). It then counts the number of lines "stopped" by each line, thus obtaining a measure of that feature's significance as an occluding contour.

The fourth and principal modification in this version of the program was a restructuring of the analysis to allow utilization of the duality results presented in the next section. Basically, what differs here is that image features are represented by the corresponding lines of sight, rather than EPI plane coordinates, and that the analysis is carried out by means of homogeneous representations of features and feature paths. This enables us to handle any camera viewing geometry as a linear problem and to eliminate the analysis of hyperbolas discussed earlier. The next section discusses this approach in more detail.

To illustrate this second version of the EPI analysis technique, we consider the outdoor scene of figure 31. Figure 32 shows the first and last im-

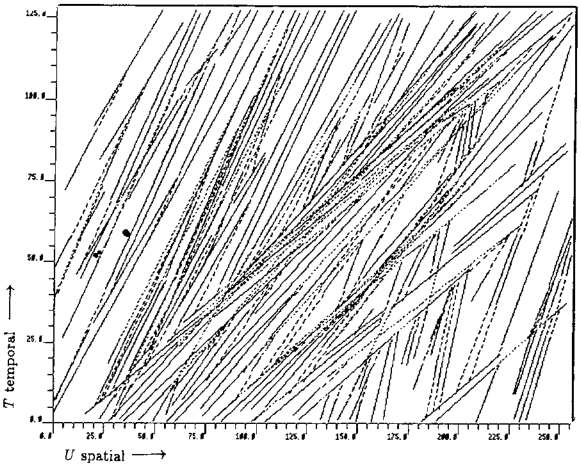


Fig. 30. Line intersections.

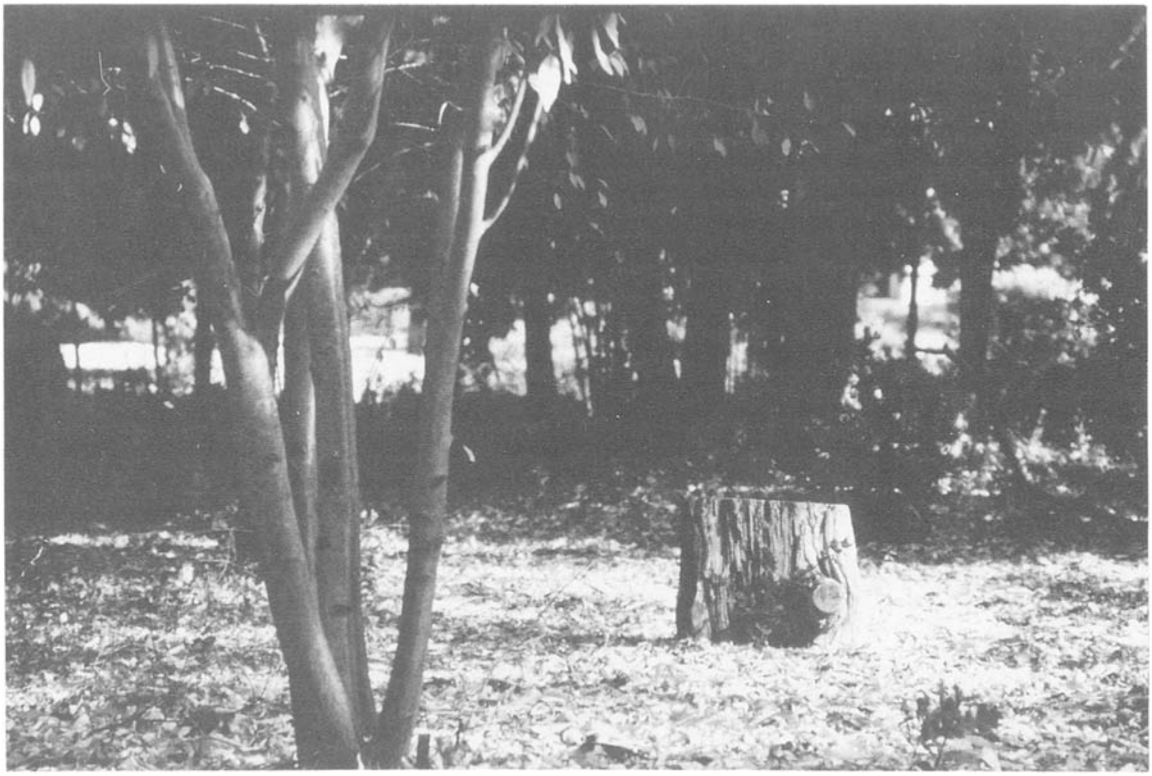


Fig. 31. Outdoor scene.

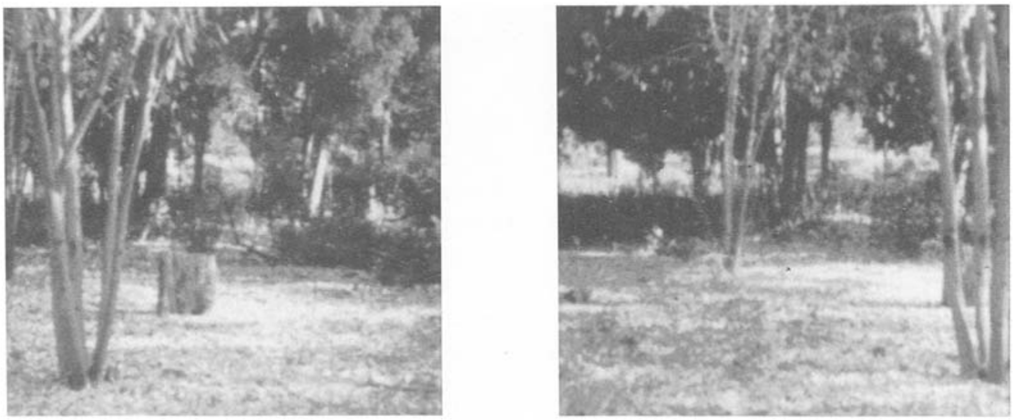


Fig. 32. First and last images in a sequence of 128.

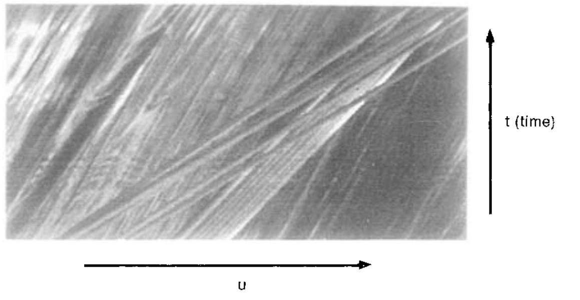


Fig. 33. EPI from outdoor scene.

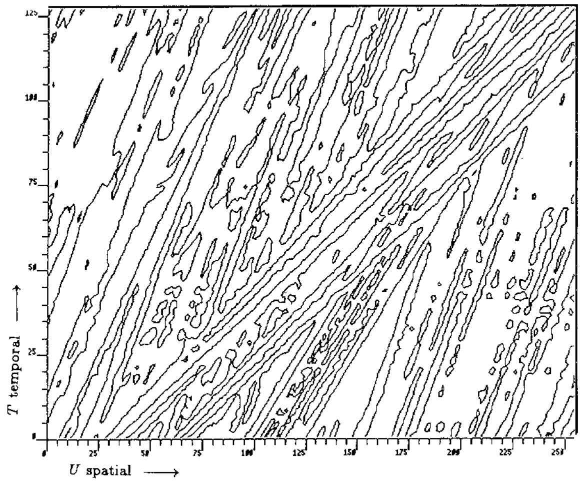


Fig. 34. Edge features in EPI.

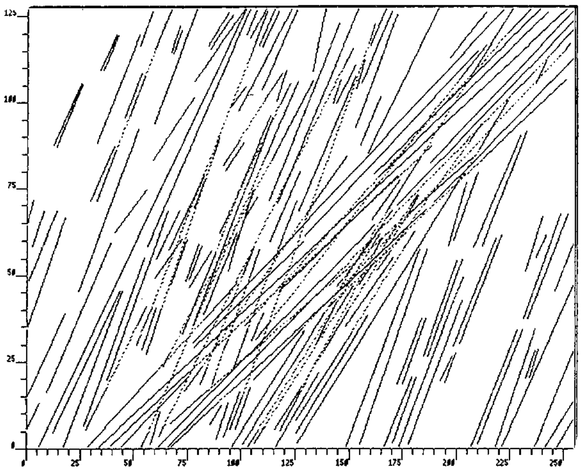


Fig. 35. Merged lines.

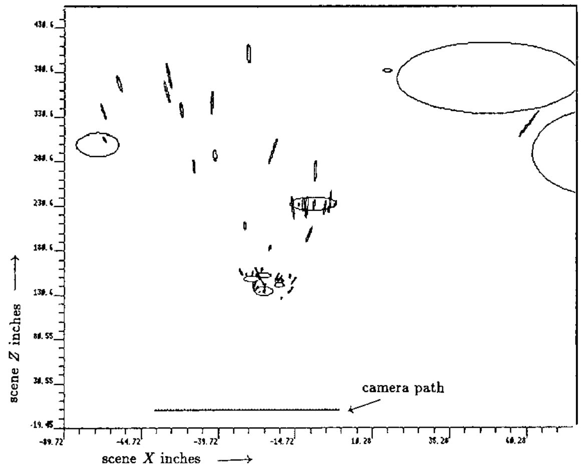


Fig. 36. *x–z* points of EPI features (*horizontal ellipses* denote approximate scene objects).

ages of a sequence of 128 taken of this scene by the SRI robot. The robot looked to its right as it moved ahead. Unfortunately, the vehicle rolled slightly on the uneven sidewalk as it advanced, so that the vertical positions of the epipolar lines changed from image to image. To compensate for this , we implemented a correlation-based tracker to follow points from one spatial image to the next. We used this to estimate the vertical shift caused by the roll and then created a new set of images that approximated the lateral motion without roll. Note that the viewing direction in this data set is again orthogonal to the camera path; the new EPI analysis handles the general case, but the process of restructuring the data by epipolar planes has not yet been fully implemented.

Figure 33 is an EPI from this sequence. Since the features are finer than those of the room scene (much of the scene consists of grass and leaves), we applied a smaller zero crossing operator to detect scene edges. Figure 34 shows the edges detected in the EPI of figure 33. Figure 35 shows the merged lines for that EPI, while figure 36 shows the estimated positions for features in the corresponding epipolar plane. The large horizontal ellipses are the approximate locations of trees and bushes, as measured with a tape, and, once again, the lower horizontal line is the camera path. Figure 37 shows a free space map, where the white region is "not free space," that is, indiscernible space. Figure 38 is another crossed-eye display, this one showing the full set of three-dimensional

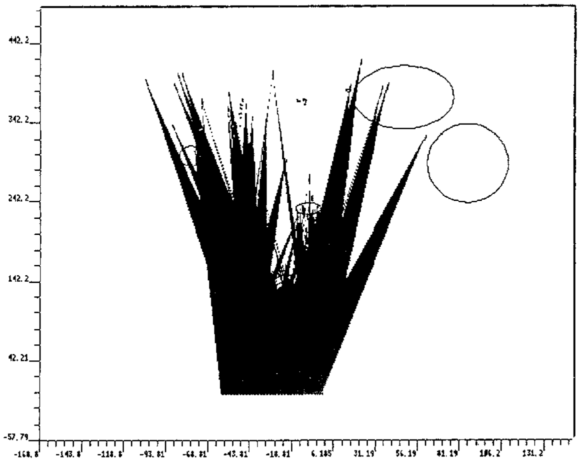


Fig. 37. Free space.

points for the two nearest trees and the stump between them; no vertical adjacency filtering has been applied for this display. Color figure 58 is a perspective view of the points established for the tree in the foreground (created with the aid of Pentland's Supersketch modeling and graphics system [70]). Each three-dimensional point is represented by a small cube colored according to its height above the ground. The intensity of each cube is a function of its distance from the camera path, with closer points being brighter. Color figure 59 shows the principal occluding contours in the scene (those that occlude more than two other features and are more than 10% nearer than those they occlude). In figure 39 (left) we show the occlusion record of a single feature at the left side of the tree in the foreground. It is seen to occlude a large number of features situated behind it. The dashed lines indicate extrapolations of the occluded features to the position (in time) when they move behind (are occluded by) the tree feature. Note that a few of the features are tracked across the occlusion. Figure 39 (right) shows a similar record for a feature from the right of the tree (as indicated by the arrow in color figure 59).

4.5 Nonperpendicular Camera Viewing Directions

The case of fixed camera orientations that are not perpendicular to the direction of motion is described by equation (7) above. Recall that a, θ_0, x_0, y_0, and f are constant, so that the left-hand side is a polynomial in u and t. When the camera orientation is perpendicular to the direction of motion, θ_0 is either 0 or π, the coefficient of the ut term is zero, and the polynomial degenerates into the line discussed earlier. Otherwise it is a hyperbola with asymptotes parallel to the coordinate axes, since in this case it can be rewritten

$$\begin{aligned}(u + f\cot\theta_0)(t + \frac{1}{a}(y_0\cot\theta_0 - x_0)) \\ = \frac{fy_0}{a\sin^2\theta_0}\end{aligned} \tag{10}$$

This form makes the locations of the asymptotes obvious. Consider figure 40. One asymptote is the line $u = -f\cot\theta_0$, which is the image location of the *epipole*, where the camera path intersects the image line. A scene point that projects to this location must lie on the x-axis, i.e., the camera path, and its image does not move. The other asymptote is the line $t = (1/a)(y_0\cot\theta_0 - x_0)$; at that time, the camera center is positioned so that

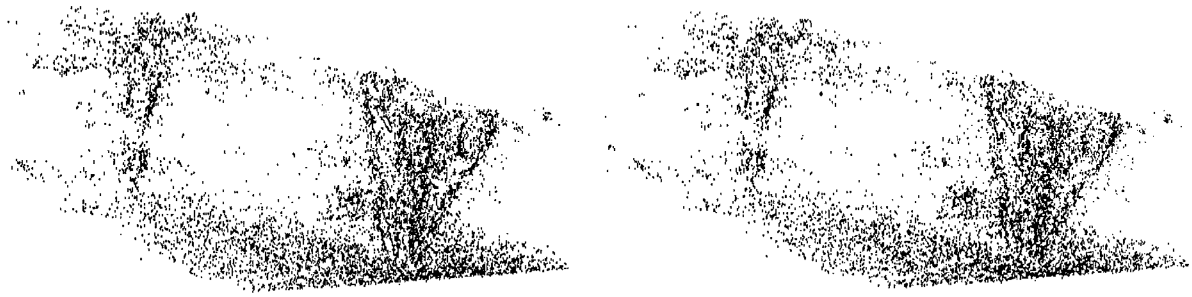


Fig. 38. Crossed-eye stereo display of scene points.

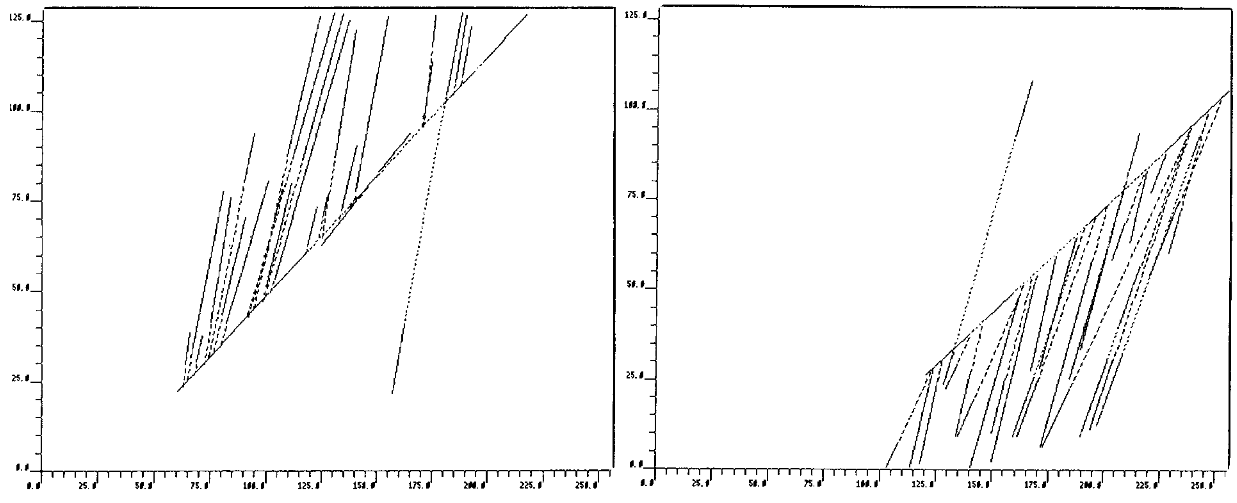


Fig. 39. Line intersections.

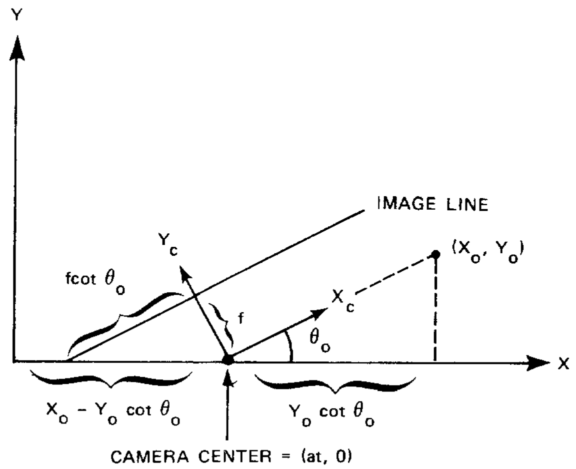


Fig. 40. Asymptotes of hyperbolic feature paths.

the line from it to the scene point is parallel to the image line, so the image of the point is formed "at infinity," (if the camera has a wide enough field of view). In the neighborhood of this location, a scene point's image coordinate becomes arbitrarily large.

4.6 Straightening Hyperbolic Paths

While the analysis of the hyperbolic feature paths arising from nonperpendicular viewing directions is fairly easy, we wanted to find a way to transform these hyperbolas into lines, which are easier to detect. Since the hyperbolas are restricted to a two-parameter family, it seemed likely that such a transformation was possible. Two approaches are suggested by Marimont [57]. In this section, we describe the first, a technique to transform the images so that they look as they would have, had the viewing direction been perpendicular to the direction of travel. In the next section, we describe the second approach, which is based on projective duality.

As an aside, we note that various authors have shown (e.g., Koenderink and Van Doorn [50] and Prazdny [73]) that the component of image mo-

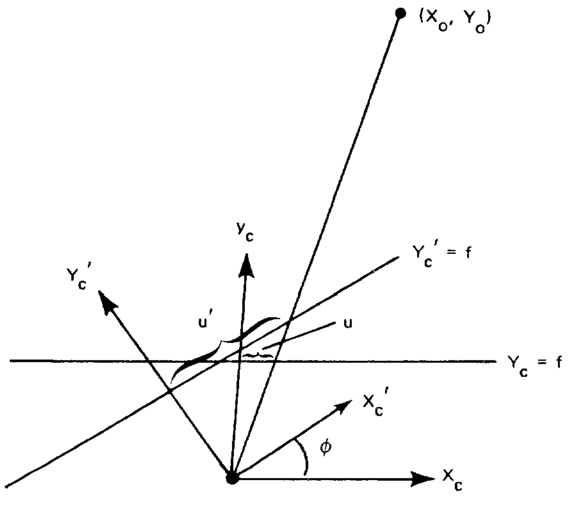


Fig. 41. Reprojecting an image point *u*.

tion caused by the camera motion's rotational component contains no information about the depths of scene points. It is therefore not surprising that it is possible to compensate for the camera's viewing direction without knowing anything about the scene.

Our algorithm for transforming the images is as follows. First we compute the equations of the lines from the camera center through specific image points, which is possible because we know the camera's position and orientation. Once we know these lines, we can intersect them with a synthetic image plane that is parallel to the direction of motion, and thus create a new image. This image is what the camera would have seen, had it been looking perpendicularly to its path (and if the effects of a finite field of view are ignored).

Consider figure 41. The camera is centered at the origin with image line $y = f$, and is looking along its y-axis. Suppose that the camera is proceeding along a line at an angle ϕ with respect to this axis. To convert the original image into one formed on an image line parallel to this line, we need to reproject the points onto the line $-(\sin\phi)x + (\cos\phi)y = f$. The line through the camera center and an image point (u, f) on the original image line has the equation $y = fx/u$. To reproject this image point onto the new line, we merely find the intersection of the line through that point with the new image line. If we let the new image coordinate be u', we find after some algebra that

$$u' = f\left(\frac{u\cos\phi + f\sin\phi}{f\cos\phi - u\sin\phi}\right) \tag{11}$$

This transformation has a singularity at $u = f\cot\phi$, which, by analogy with the t-asymptote of the hyperbola in equation (10), is that value of u for which the line of sight is parallel to the second image line; the line of sight thus has no finite intersection with the second image line. It follows that the line of sight to a point on the camera path has no intersection with the new image line. Since it would never move in the new image, the singularity results in no loss of information.

Thus, if the orientation of the camera coordinate system is θ_0, we can always linearize the feature paths by letting $\phi = -\theta_0$ and transforming the images by means of equation (11). Figure 42 (top) illustrates this process for a simulated planar scene. On the left is the scene containing a few polygonal objects and a straight camera path. The short line segments splitting off from the path indicate the camera's viewing direction, while the numbers to the right are the "times" at which the camera was at that point on the path. In the center are the hyperbolic feature paths, with the times labeling the vertical axis. On the right are the straightened feature paths.

It may seem that little is gained through this transformation, since the same two parameters—the plane coordinates of the scene point—determine both the hyperbola and the line. Even so, lines are far simpler to deal with, both analytically and computationally.

Perhaps even more importantly, the linearizing transformation is applicable even when θ_0 varies with time, since each image can be transformed independently with a ϕ that varies with time as well. When θ_0 varies, there is no single hyperbola or line that can be fitted to an image feature path and from whose parameters the location of the corresponding scene point can be inferred. See figure 42 (bottom) for an example; the figure is basically the same as figure 42 (top), except that the line segments in the former indicate that the camera's viewing direction varies along the path.

The situation in three dimensions is analogous and will not be discussed here. The derivation can be found in Marimont [57]; Kanatani [45] independently obtains similar results.

4.7 Discussion

To date, we have processed three image sequences of complex scenes (a partial analysis of the third is presented in section 5.6). The success of the feature tracking, even in the areas of grassy texture in the foreground of the outdoor sequence, suggests that the technique is robust. The processing is basically identical for the different data sets; only the selection of the space constants for the convolutions requires manual intervention (the finer texture of the outdoor scene called for smaller Gaussians). Selecting the appropriate scale for analysis is a difficult problem in many areas of vision; in our case, performing the analy-

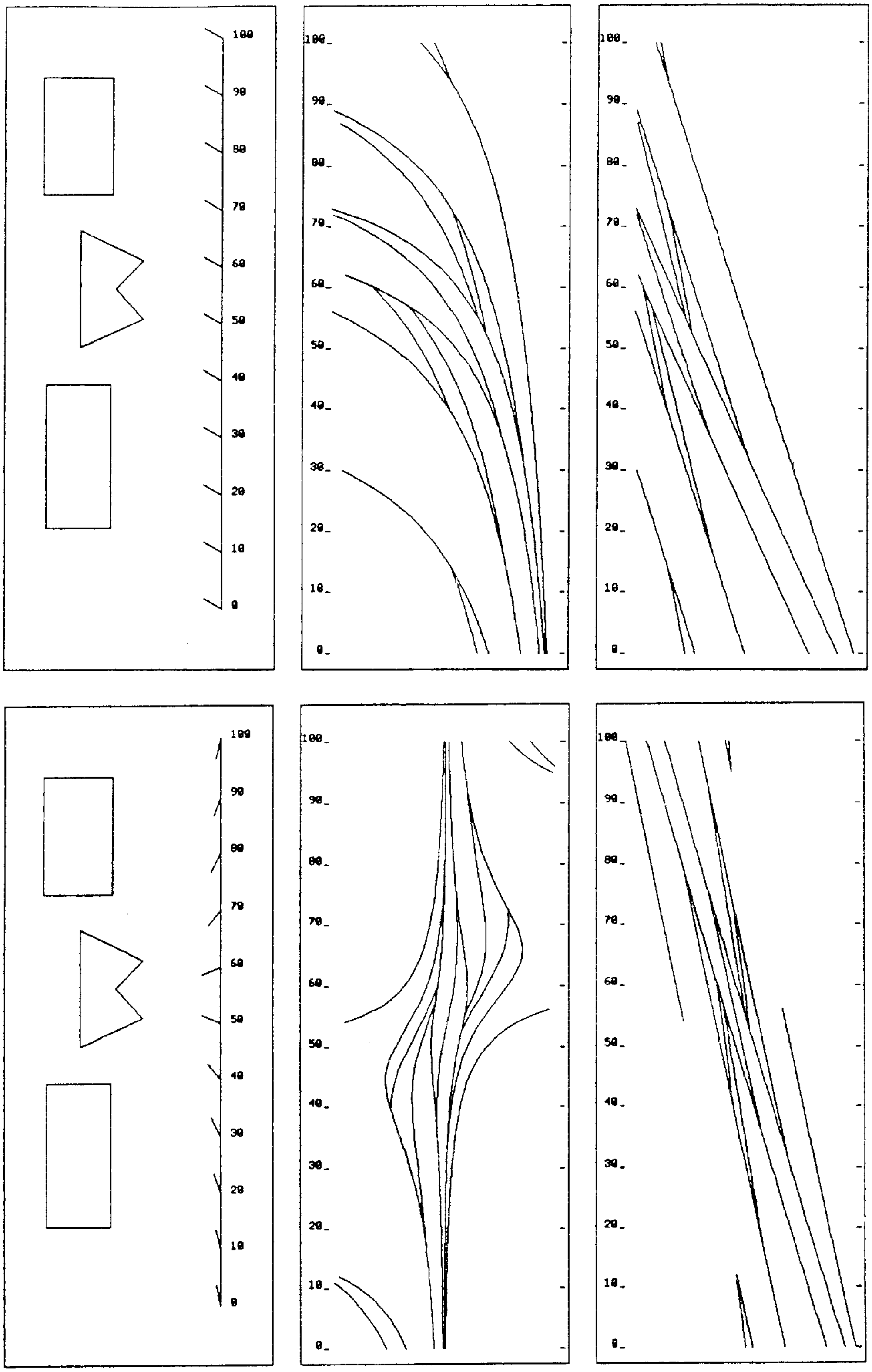


Fig. 42. Reprojecting hyperbolic feature paths for camera viewing direction: fixed (*top*), varying (*bottom*).

sis at several scales simultaneously may resolve the issue.

The linearity of feature paths is an obvious advantage of this approach over traditional correspondence-based approaches. Matching elsewhere is dependent upon the local appearance of a feature. If the feature's appearance differs significantly between views, which can happen when it is seen against different backgrounds or near an occluding contour, it is unlikely to be matched correctly. With our approach, the feature's location in space, determined by the parameters of its linear path, is the principal measure for merging observations; its local appearance plays less of a role. This means that we can collect observations of a feature wherever it is visible in the scene, even if it is seen against different backgrounds and under different illumination.

The linear motion constraint at the heart of EPI analysis can also be used to constrain the search required to track features from one image to the next. In this approach, the first step is to select distinctive features to be tracked; the second is to track them. Just as in EPI analysis, the epipolar planes defined by the linear motion constrain the feature matches to lie along epipolar lines, thereby reducing the search to one dimension. In addition, if the scene is static, locating a feature in a second image is sufficient to compute the three-dimensional coordinates of the corresponding scene feature. Once the feature's three-dimensional position has been computed, it is possible to predict its locations in all the images in the sequence. These predictions can reduce the search further by restricting it to a small segment of the epipolar line. Finally, when a scene feature has been located in three or more images, a fitting procedure can be applied to improve the estimate of its three-dimensional position, which in turn can reduce the size of the segment to be searched.

The difference between this approach and EPI analysis is that, in EPI analysis, the acquisition and tracking steps are marged into one step: finding lines in EPIs. Locating a line in an EPI is equivalent to acquiring a feature and tracking it through many images. Therefore, in addition to simplifying the matching process, EPI analysis provides the maximum amount of information to the fitting process that computes the three-dimensional location of the associated scene feature. Finally, EPIs encode occlusion events in a direct pictorial way that simplifies their detection and interpretation.

One difficulty with EPI analysis is that it produces depth measures only for those features with a component perpendicular to their epipolar plane. A feature that runs along the plane, like the horizontal rail running across the lower middle of the scene in figure 6, may be detected at its endpoints, but not elsewhere. The current effort to build a spatiotemporal surface description will enable us to represent these horizontal structural elements.

5 The Relevance of Projective Duality

Our approach to the analysis of image sequences has a number of properties that are desirable in a more general scene-analysis technique. The class of camera motions to which it applies directly—straight-line translation with the viewing direction fixed—produces a sequence of images whose analysis can be decomposed into the analysis of separate EPIs, each of which contains information about a different planar slice of the scene. The structure of each EPI is both simple and informative: the image point paths are either linear or hyperbolic, and are simply related to the locations of the corresponding scene points. The occlusions and emergences into visiblity of scene points as the camera moves are given by the topology of the image point paths. Moreover, as we showed at the end of the previous section, it is possible to transform imagery taken with a varying camera viewing direction so that all the fixed viewing direction results will be valid.

Still, the technique, in its present form, applies only when the camera motion is linear and only to stationary point objects. In this section, we use concepts from projective duality to extend the technique to more general camera paths, curved objects, and independently moving objects. First, using duality in the projective plane, we treat the case of planar camera motion. A sequence of images collected by a mobile robot, moving along a planar but nonlinear path, is analyzed to reconstruct the planar slice of the scene that contains the camera motion. Finally, using duality in projective space, we outline some generalizations of these results to the three-dimensional world.

Others have applied projective duality to problems in machine vision, although we appear to be the first to use it in motion. For example, Huffman [38,39] and Mackworth [56] deal with interpreting line drawings of polyhedra, while Roach and Wright [77] consider problems in manipulating polyhedra.

5.1 Planar Camera Motion

In the previous section, we introduced the planar world because linear camera motion makes it possible to decompose a spatial scene into planar slices that can be reconstructed independently. Now we study the planar world both for its own sake and because the techniques applicable to the planar world have analogues for the spatial world.

Here we use duality in the projective plane to extend EPI analysis to arbitrary camera motions, curved objects, and moving objects in a planar world. These planar techniques can be used to reconstruct only that portion of the scene that lies in the plane of the camera motion. This is because it is the only epipolar plane shared by every pair of locations along a nonlinear but planar camera path. Full three-dimensional scene reconstruction and arbitrary three-dimensional camera motion require duality in projective space.

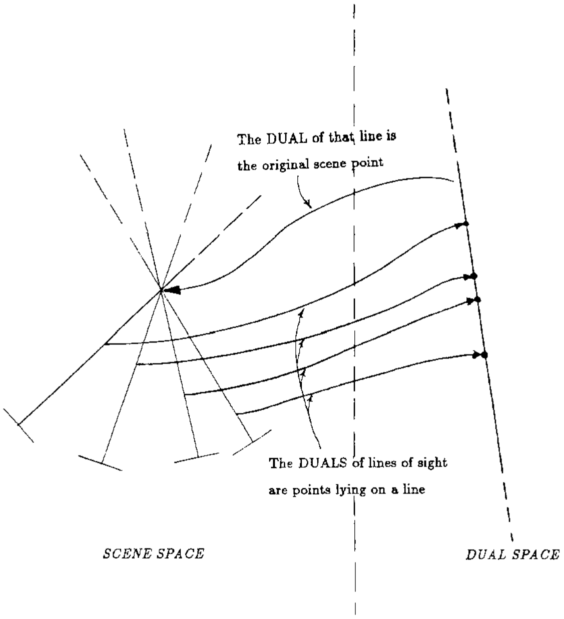
Consider the motion of a camera around a point in a planar scene; let us assume for the moment that the point is always in the camera's view. As the camera moves, the lines of sight from the camera center to the point are a subset of the pencil of lines through the scene point. At each point on the camera's path, if the camera's position and orientation are known, the parameters of the line of sight can be computed from the image coordinate of the projection of the scene point and the camera center. Now let us consider a stationary camera viewing several points in a scene. The lines of sight from the camera center to the scene points are a subset of the pencil of lines through the camera center. Here, too, if the camera's position and orientation are known, the parameters of each line of sight can be computed.

The principle of duality in the projective plane states that any axiom or theorem remains true when the word "point" is interchanged with "line" (and, as appropriate, certain words like join and meet, collinear and concurrent, vertex and side) [20]. For example, two points determine a line, so that, by the duality principle, two lines intersect in a point. A point is said to be *dual* to a line, and vice versa. In particular, the point with homogeneous coordinates (a, b, c) is dual to the line which satisfies the equation $ax_1 + bx_2 + cx_3 = 0$; the line is said to have homogeneous *line* coordinates (a, b, c).

All lines that intersect at **p** are dual to points that lie on the line **l** dual to **p**. Moreover, if we add the point at infinity to the line **l**, so that the line is topologically equivalent to the circle (since it "wraps around" at the point at infinity), the ordering of the lines through **p** induced by their orientation is the same as the linear ordering of their duals along **l**.

Now let a scene point be **p** and the lines of sight from a camera path to **p** be $\mathbf{l}_i$. The duals of these lines of sight must be points that lie on the line dual to **p** *no matter what the camera path*. As the lines of sight sweep out a "wedge" centered at **p**, the duals of the lines of sight trace out a portion of the line dual to **p**, as illustrated in figure 43.

If the camera path is known, the homogeneous parameters describing the lines of sight can be computed and the point dual to each line found.



The DUAL of that line is the original scene point
The DUALS of lines of sight are points lying on a line
SCENE SPACE
DUAL SPACE

Fig. 43. Points dual to lines of sight through a scene point.

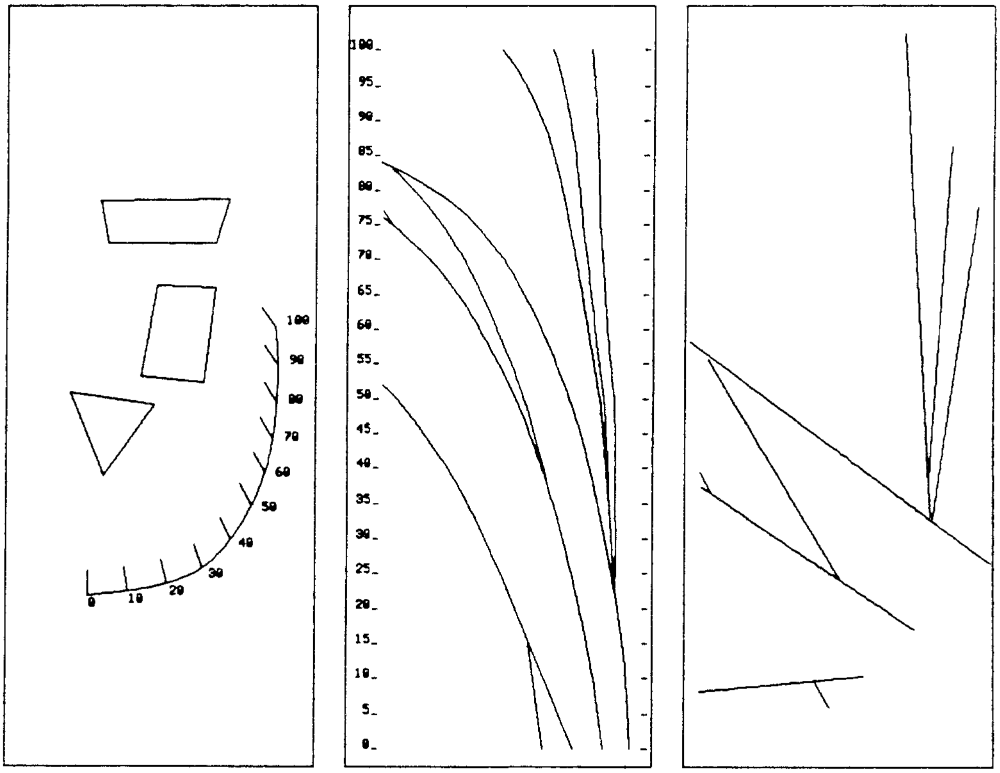


Fig. 44. A planar scene (*left*); the image history (*center*); the dual scene (*right*).

Then a line can be fitted to these dual points. Since these points are dual to the lines through the scene point, the fitted line is dual to the scene point. Thus, the parameters of the fitted line give the homogeneous coordinates of the scene point. This is quite similar to the analysis of a line in an EPI that we discussed in the preceding section, except that there the line appears in the image, whereas here it must be computed from the duals of the lines of sight.

The dual to the camera center also has a simple geometric interpretation. When the camera can see several points in the scene at once, its lines of sight to them all pass through the camera center. These lines are dual to points that must lie on the line dual to the camera center. As the camera moves, the duals to the lines of sight to each scene point trace out a line segment, but the trace proceeds along a line that is dual to the camera center. Each image's "contribution" consists of those points that are dual to the lines of sight. But, at any given image, all these lines of sight pass through the camera center. That image's contribution must therefore lie along the line dual to the camera center.

Although the linearity induced by duality surprised us initially, it means merely that solving for the location of a scene point is a linear problem if the camera motion is known. This is obvious (in retrospect), since in that case it amounts to finding the point at which all the lines of sight intersect. The points dual to the lines of sight form a picture of this linear problem. Our goal in this section is to show that these dual structures are a generalization of those we have already analyzed in EPIs.

Figure 44 depicts a simulated planar scene (left), the image history (center), and the "dual scene" (right), the duals of the lines of sight. The scene contains several polygonal objects and a curved camera path, jutting out from which are short line segments that indicate the camera's viewing direction at that point. The numbers next to the path indicate the time at which the camera passed that point. Throughout this article, the units of time chosen are identical to the number of images formed up to that time; in this scene, the camera forms a total of 100 images.

The column of numbers at the left of the image history indicates which image is represented along that horizontal line; consequently, the first image is at the bottom and the last one at the top. Note that the topology of the dual scene is identical to that of the image history, but that in the dual scene all paths are linear. Each such path corresponds to a single scene point whose location can easily be estimated by fitting a line to the path; the scene point is the dual of that line.

5.2 Collineations

Some of the figures in this section involving points dual to the lines of sight have had collineations applied to them to make them easier to interpret. (Recall that a collineation is a nonsingular linear transformation of the projective plane and can be represented as a 3 × 3 matrix.) This is often necessary for two reasons: first, a collineation can have the effect of a generalized scaling, which can increase resolution in the areas of interest; second, points "near" the line at infinity can be arbitrarily far away from the origin and thus hard to draw in a finite area. Fortunately, collineations leave the properties of those points that are of interest to us here invariant—most importantly topology and collinearity. And even though the coordinates of a collineated point change, since a collineation is invertible, the original coordinates can always be recovered if the collineation is known. Thus, while a collineation can change the appearance of a figure, the analyses of the figures presented here either remain the same or require only trivial modification.

5.3 Occlusions and Disocclusions

The situation with occlusions and emergences into visibility of points in the scene is similar to that in EPI analysis, but must be somewhat generalized. We saw earlier that a merge of two image paths in an EPI corresponded to a merge of the corresponding lines of sight, which in turn corresponded to an occlusion. A split of one image path in an EPI into two corresponded to a split of the corresponding lines of sight, which in turn corresponded to an emergence into visibility.

The situation with the duals to the lines of sight is even simpler, since each point in the dual is dual to a line of sight. A line of sight whose parameters are continuously changing gives rise to a dual curve whose points are likewise continuously changing. Two lines of sight whose parameters are continuously changing and that ultimately coincide at an occlusion give rise to two curves that ultimately meet, so an occlusion corresponds to a merge in the duals of the lines of sight. A similar argument shows that a split in the dual corresponds to an emergence into visibility. Loosely speaking, we rely here on a suitably defined topological equivalence between lines of sight in the scene and the points to which they are dual.

The *orientations* of the dual lines at splits and merges are not related quite as simply to one another as before. Consider the situation at a split or merge illustrated by figure 45. The camera center **a** and two scene points **b** and **c** determine a line **l**, with **b** closer to the camera center than **c**. The duals of the three points are lines **a**′, **b**′, and **c**′, which pass through the point **l**′ dual to **l**.

Recall that the ordering on the lines through a point induced by their orientation is the same as the order of the points to which they are dual along a line if the point on the line at infinity is included to make the line topologically equivalent to a circle. Note that every point at infinity $(p_1, p_2, 0)$ is dual to a line through the origin $p_1x + p_2y$

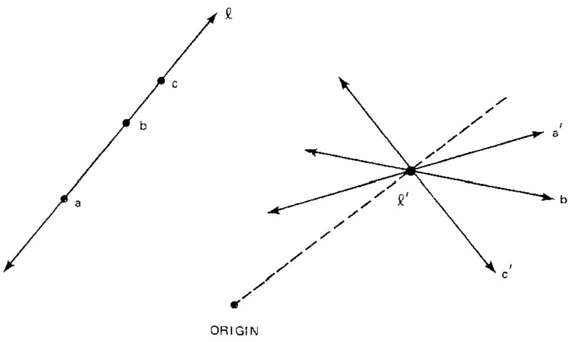


Fig. 45. The camera center and two scene points at a split or merge (*left*); the dual scene (*right*).

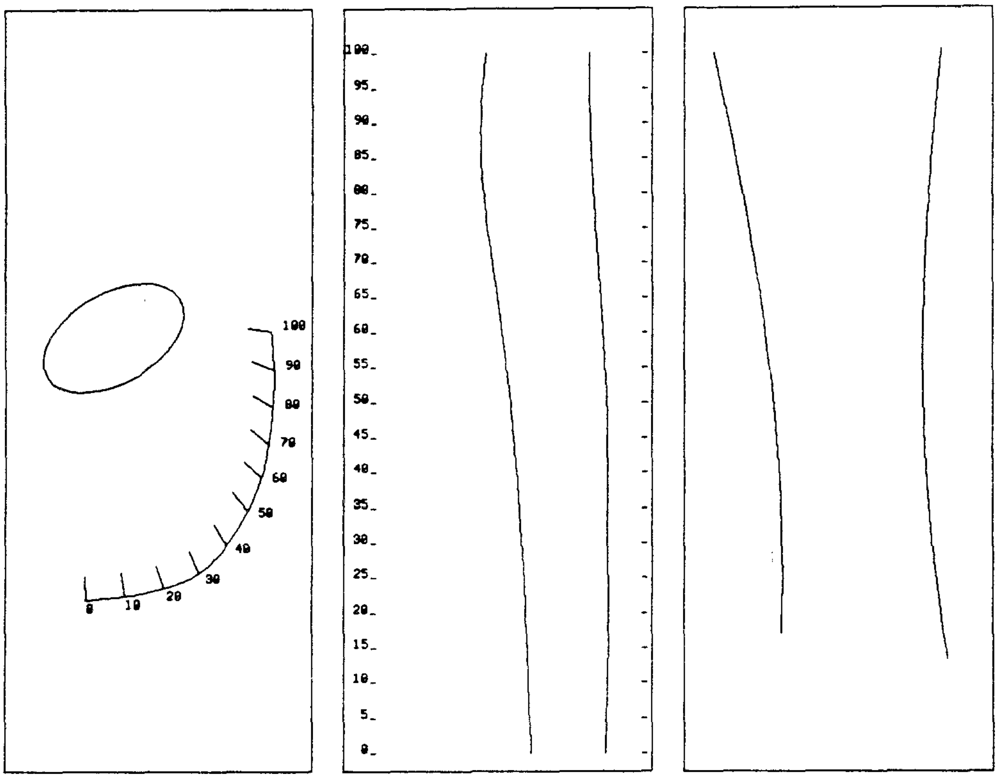


Fig. 46. A scene with a curved object (*left*); the image history (*center*); the dual scene (*right*).

= 0. The point on **l** at infinity must be dual to the line through the origin and **l**′.

If we begin at **a** and move along **l** in the direction of **b** and **c**, we pass through points that are dual first to **a**′, then to the lines through **l**′ rotated from **a**′ in the direction of **b**′. Moving along **l** in the other direction means rotating from **a**′ in the other direction, so that, just as the point at infinity is encountered before **b** and **c**, the line through the origin will be encountered before **b**′ and **c**′.

Our goal is to establish a rule specifying which of the two lines at an intersection (both of which are dual to a scene point) "cuts off" or "spawns" the other at a split or merge. This is equivalent to knowing which scene point is closer. From the above discussion, it is clear that, when **a**′, **b**′, and **c**′ are given, whichever line is encountered first when **a**′ is rotated away from the line through the origin is dual to the scene point that is closer to the camera center.

The one exception to this rule occurs when the dual scene is drawn after some collineation has been applied to the lines. In this case, we replace the line through the origin with one that passes through the point dual to the line at infinity.

5.4 Curved Objects

In the plane, a curved object is simply a curve. Image edges are formed at the projections of those points on the curve that have lines of sight tangent to the curve; such edges are called limbs. As the camera center moves around the object, the lines of sight trace out the tangent envelope of the curve, and the scene point corresponding to the image edge point actually moves along the curve. Because the imaged point is not stationary, the analysis of image paths above is not applicable.

Fortunately, the dual of a curve has a simple and useful interpretation in this context. The dual of a differentiable curve is the dual of its tangent envelope. That is, at any point on the curve, there is a line tangent to the curve at that point. The dual to the curve at that point is the dual to that line. For example, it is well known that the dual of a conic is also a conic. The equation of a conic can be written $\mathbf{x}^tA\mathbf{x} = 0$, where $\mathbf{x} = (x_1, x_2, x_3)^t$ *and* $\mathbf{A}$ is a 3×3 matrix. The tangent envelope of this conic is the set of lines that satisfies $\mathbf{l}^tA^{-1}\mathbf{l} = 0$, where $\mathbf{l} = (l_1, l_2, l_3)^t$ and represents the line that satisfies $l_1x_1 + l_2x_2 + l_3x_3 = 0$. The dual of the conic $\mathbf{A}$ is the set of points dual to the lines that make up its tangent envelope $\mathbf{A}^{-1}$; this set of points is obviously also a conic.

Because the image edge path corresponding to the limb of a curved object arises from the lines of sight that trace out the tangent envelope of the curve, the dual of the lines of sight must therefore be the dual of the curve. In general this dual will not be a straight line. To recover the shape of the original object, we fit a curve to the dual of the lines of sight and take the dual of this fitted curve—which is, of course, the original curve. Note the similarity to the case of stationary points. There the dual of the image edge path is a straight line. To recover the location of a stationary point, we fit a line to the dual of the lines of sight and take the dual of this fitted line—which is the original point. One advantange of this approach is that it is not necessary to distinguish between image edge paths corresponding to stationary points and those that correspond to curved objects, since in the dual scene the type of curves that arise makes the distinction obvious.

Such an approach could be applied to the situation depicted in figure 46. The scene consists of an ellipse viewed from points on a curved path. The dual scene, computed from the camera path and image history, shows part of the hyperbola that is dual to the ellipse. To recover the part of the ellipse that was actually viewed, we must fit a curve to the dual scene data; the dual to the fitted curve is the part of the ellipse we seek.

5.4.1 Occlusions and Disocclusions. The occlusions and disocclusions of curved objects have the same branching structure or topology as that of polygonal objects. This is because the number and continuity of the lines of sight are the same in both cases, and the topology of the dual is identical to the topology of the lines of sight. The other issue in the interpretation of occlusions and disocclusions is the orientation of the image paths in the dual. In the case of stationary scene points, the orientation is directly linked to the locations of the points relative to the camera and thus makes it possible to predict which image path would be terminated or spawned at a merge or split.

Since the dual of a curve is the dual of its tangent envelope, the dual of the dual's tangent envelope must be the original curve. That is, at any point on the dual, the dual of the line tangent to that point is a point on the original curve. This means that at a split or merge involving two curved objects, the tangents to the dual curves are dual to the points on the objects involved in the occlusion or disocclusion. Thus the situation locally is identical to that when two polygonal objects are involved, since the tangents to the image paths there are dual to the scene points involved also; the only difference is that in this case the tangents are constant over the entire path. The analysis of the orientations of the paths for stationary scene points therefore is applicable.

5.5 Moving Objects

In this section, we consider objects that are moving independently of the camera. Suppose, for the moment, that the object consists of a single point,

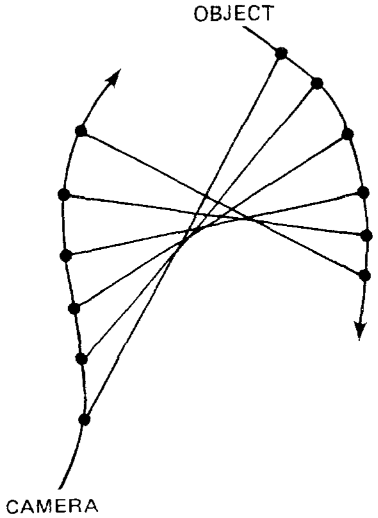


Fig. 47. A moving point object.

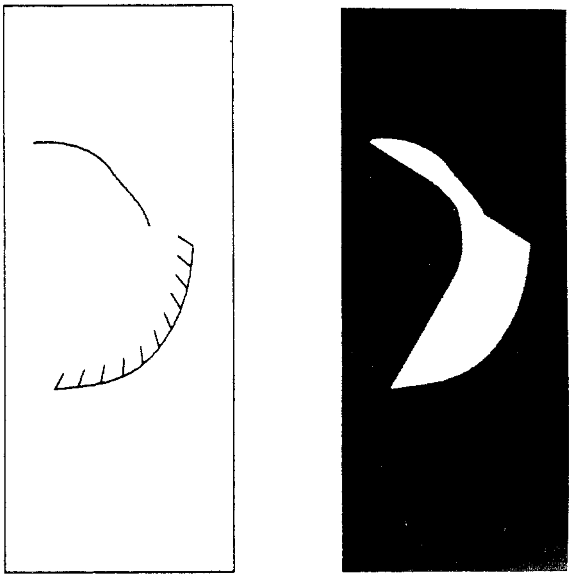


Fig. 48. A moving object and the free-space map.

as in figure 47. The lines of sight from the camera to the object are known, as usual, and, if the motion of the camera and the object satisfy certain smoothness conditions, their dual is a continuous curve. This dual corresponds to the tangent envelope described by the lines of sight, except that, unlike the case of curved objects, whose curve is formed by the lines of sight, the tangent envelope does not correspond directly to a scene object.

Let us imagine, however, that the scene contains a curved object with the same tangent envelope. Since this implies that the dual would be unchanged, the image history would not change either. That is, the motions of the point-object and the camera define a unique *equivalent stationary curved object* (*ESCO*) that could replace the moving object in the scene and result (locally, at least) in exactly the same image edge path. (The reason for the restriction of locality is that the ESCO may be self-occluding when considered as a whole, so that it would have to be transparent and still yield the appropriate limbs to produce an identical image history.)

Because the lines of sight to the moving object are tangent to the ESCO, and, furthermore, the free space induced by the moving object has these lines of sight as one boundary, the tangent envelope of that boundary of the free space is, under certain conditions, the same as that of the ESCO. In figure 48, the outline of the ESCO, which in this case is the curved, leftmost boundary of the free space, can be seen quite clearly.

Since every moving object has a unique interpretation as an ESCO, it is important to ask whether other attributes of the dual scene can be used to distinguish between moving and stationary objects. If a rigidly moving object gives rise to several image edge paths, it may be possible to determine whether the ESCOs should be uniquely interpreted as a rapidly moving object. The existence of one rigid motion that explains all the image paths is powerful evidence that one rigidly moving object, rather than a group of unrelated curved objects, is in the scene. As this level of interpretation is somewhat higher than the others we have been considering, we shall not pursue it here.

The other approach to distinguishing between moving and stationary objects is to use occlusions and disocclusions, if they are available. Because a moving object can give the impression of being a stationary curved one, it is possible that the evidence in the dual scene of the occlusions and disocclusions connected with the moving object has no consistent interpretation. For example, suppose that a moving object is being occluded by a stationary one, and that the moving object's ESCO is closer to the camera than the stationary object. The occlusion in the dual scene will show, on the basis of local evidence alone, that a farther object is occluding a closer one; therefore, a moving object must be involved.

This situation is depicted in figure 49. The scene shows a triangle moving in the direction indicated by the arrows between a larger triangle and the camera. The two "snapshots" of the moving triangle are taken at time 27, as the lower left vertex of the larger triangle becomes visible, and at time 89, as the upper right vertex of the larger triangle is occluded. Note the corresponding split and merge in the image history. In the dual scene, the lines corresponding to the camera centers are approximately vertical and move from right to left. The rightmost dashed line corresponds to time 27, the leftmost to 89—so that these lines intersect the split and merge, preserving the topology of the image history.

The small black square at the lower right of the dual scene marks the point dual to the line at

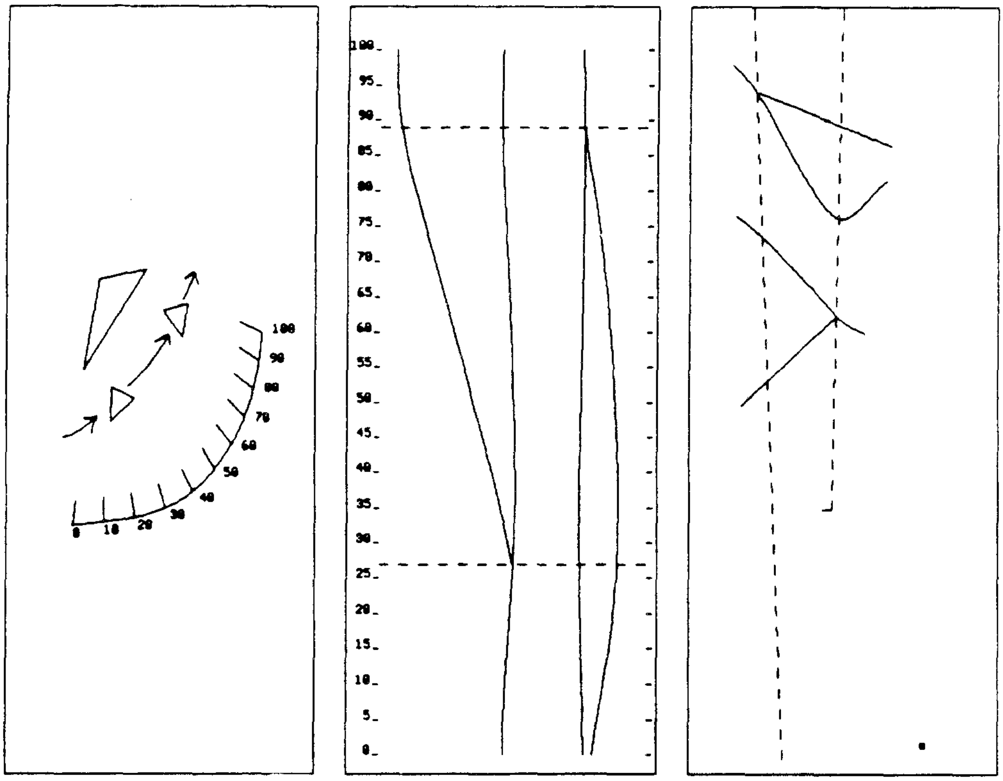


Fig. 49. A moving object and an inconsistent occlusion.

infinity. We want to determine which of the two points corresponding to the paths that split at time 27 is closer to the camera center, so we use the method discussed in an earlier section. The camera center dual line at time 27 must be rotated clockwise to coincide with one of the tangents to two paths before coinciding with the line determined by the split and the point dual to the line at infinity. Thus, it coincides first with the line spawned by the split, and the corresponding scene point must be closer than the one that corresponds to the tangent to the other path. But a split marks an emergence into visibility, so the path spawned by the split must always be dual to the farther scene point. We thus have a contradiction, which in this case arises because the other path corresponds to a moving scene point.

Now we determine which of the two points corresponding to the paths that merge at time 89 is closer to the camera center. Again the camera center dual line at time 89 must be rotated clockwise. This time it coincides first with the path terminated by the merge, so the point dual to this line must be closer to the camera. Since a merge is an occlusion, however, the path terminated by the merge must always be dual to the farther scene point. Again a contradiction arises because the other path corresponds to a moving scene point.

5.6 Experimental Results

In this section, we describe the results of some experiments designed to test the methods proposed for analyzing images induced by planar camera motion. A sequence of 128 images was collected by the SRI mobile robot as it moved along a circular path with its camera viewing direction fixed to be perpendicular and to the left of its direction of motion, so that the camera was looking toward the center of the circle. At that center stood a table on which there was a collection of typical office objects: some books, a computer terminal, a briefcase, a cup, and a

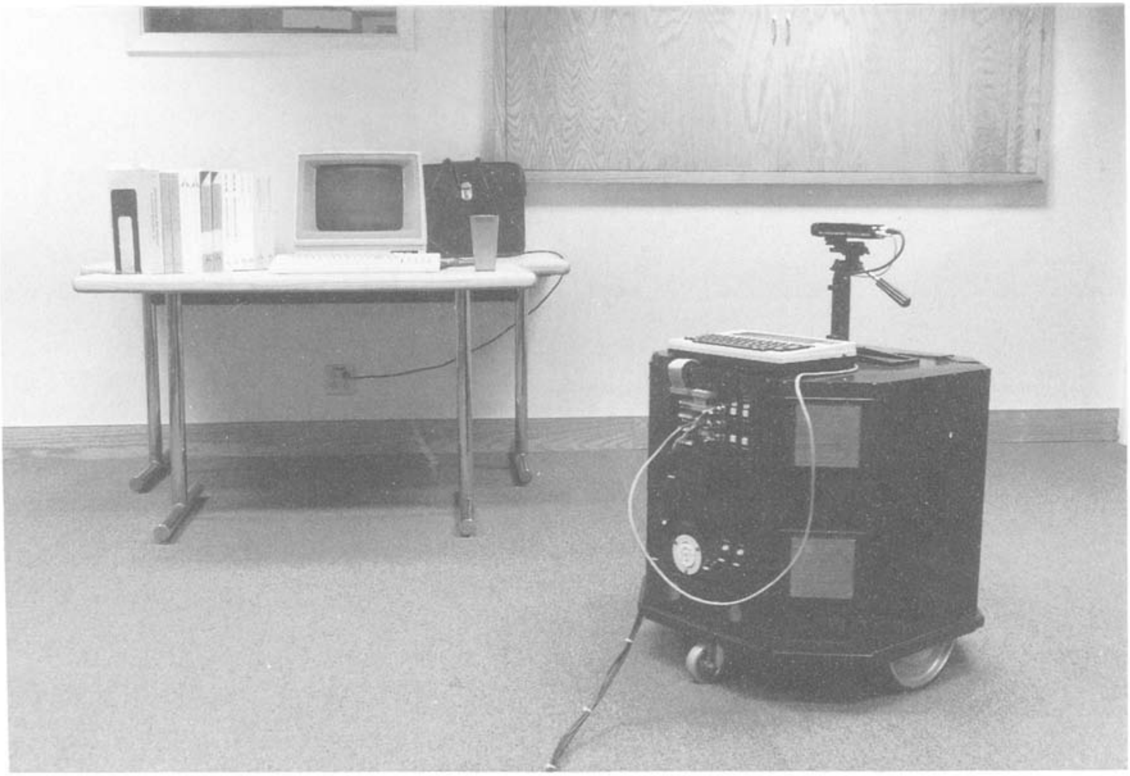


Fig. 50. The SRI mobile robot collecting images.

pumpkin.[2] The path covered about 135 degrees of arc, and at its midpoint the robot was directly in front of the table. Each image contains 240 scanlines by 256 pixels. Figure 50 shows the scene as the robot was moving around; figure 51 (top) displays several frames from the acquired sequence.

Since the camera's motion is planar and its viewing direction is fixed, the techniques discussed above for inferring the structure of planar scenes apply to the scanline in the sequence contained in the epipolar plane that coincides with the plane of the camera motion. In this case, it is the middle scanline, no. 120. Figure 51 (center) shows a slice of the images through this scanline; that is, the slice is an image of 128 rows, each consisting of 256 pixels. The i^{th} row of this EPI (counting from the bottom) is the 120th scan line of the i^{th} image.

The methods employed to analyze the image sequence require that the camera's position and orientation at each image be known. In these experiments, the robot's onboard dead-reckoning system was the only source of this information. This so-called "trajectory integrator" combines readings from shaft encoders on the two independently controlled wheels to compute a running estimate of the robot's position and orientation.

Once the trajectory information has been obtained, it is possible to transform the intensities of the EPI in figure 51 (center) so that edges in the scanline image that correspond to stationary scene points become straight in the "dual image." This is because each [rectangular] pixel in the EPI maps to some quadrilateral in the dual scene that is painted (after some resampling to avoid aliasing) with the pixel's intensity. Figure 51 (bottom)

[2] Typical, that is, around Halloween, at which time the experiment was performed.

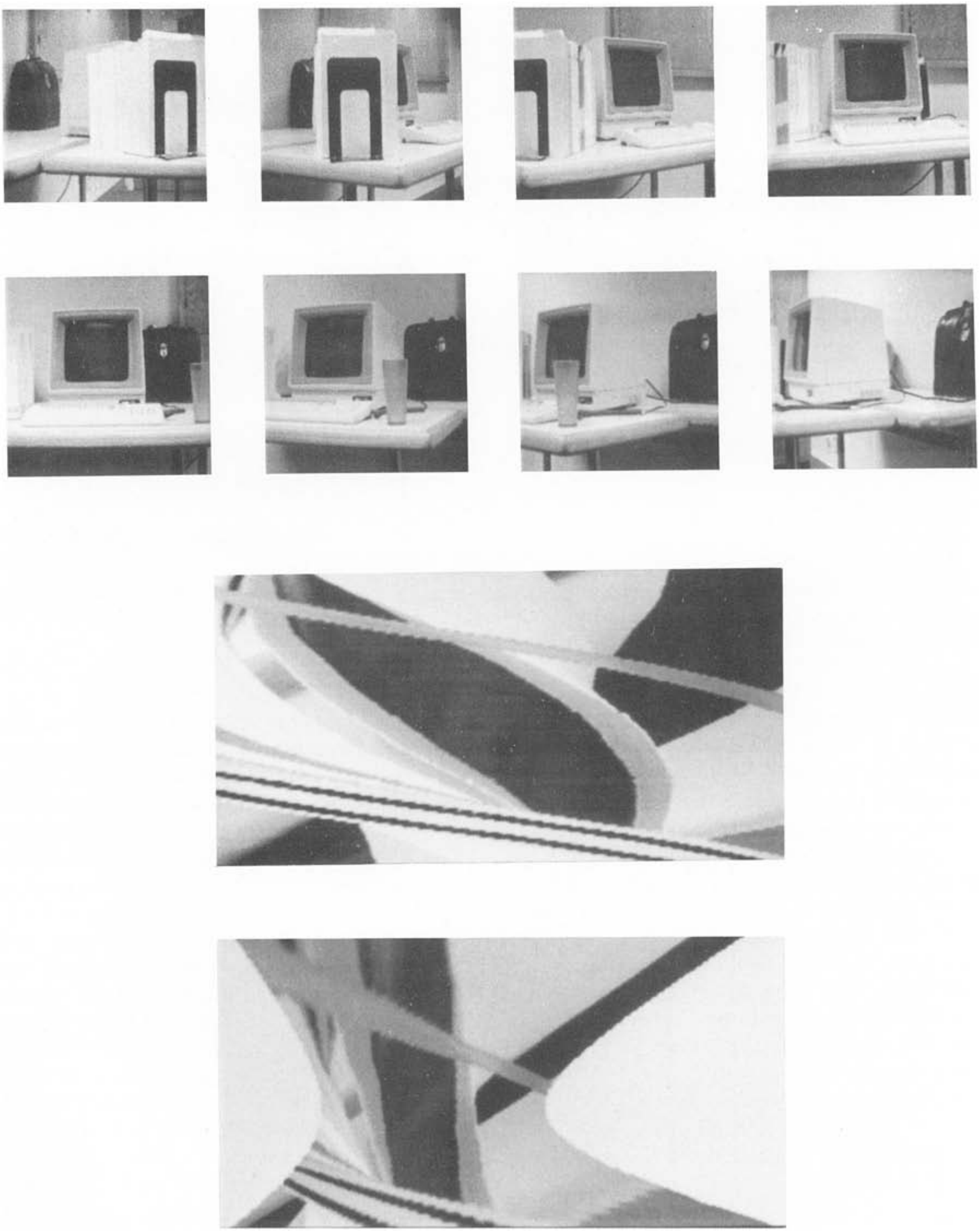


Fig. 51. Frames from the image sequence collected by the mobile robot (*top*); the EPI from the middle scan line (*center*); the EPI transformed to straighten edges (*bottom*).

shows the result, after the first and last few images have been clipped to increase the resolution in the rows of the transformed image corresponding to the middle of the image sequence. Note that most of the curves in figure 51 (center) are now straight, although there are still small wiggles in places (probably because of noise in the trajectory integrator).

The positions of the objects on the table were measured manually to provide a standard for reconstructing the scene. From these manual measurements, a model of the planar slice of the scene imaged by the middle scanline was constructed. The robot's path, as reported by the trajectory integrator, was added; the resulting planar scene is depicted in figure 52 (left), where the point features giving rise to edges in the simulator are marked with small black squares (i.e., the limbs of curves are unmarked). To test the fidelity of the simulated scene, we then simulated its imaging along the camera path; the result is figure 52 (right). Only the vertices of polygons and the limbs of curved objects give rise to image edges in the simulated EPI. There are a few minor differences between the real scanline image in figure 51 (center) and the simulation.

In the experiments that were performed, the location of scene points corresponding to image edge point paths in the EPI had to be estimated from the duals of the paths. There are two ways to compute these dual paths. Either the image edge point paths can be detected in the original EPI and their duals computed by using the camera path, or the duals of such paths can be detected directly in the transformed imagery. In noise-free images, these two approaches are for most practical purposes equivalent, but, because the transformed intensities constitute a variable stretching and shrinking of the original image, where the statistics of the noise are fairly uniform, the comparable statistics in the transformed image vary from pixel to pixel. This makes it difficult to detect edge points. For this reason, we choose here to perform edge detection only in the original, untransformed imagery.

To analyze the EPI of figure 51 (center), the program used in analyzing the linear-camera-path image sequences discussed above was modified to incorporate the theories introduced in this section. The steps in the analysis are as follows: (1) detect edge points, (2) link edge points into lists (called ledgels), (3) break ledgels so no ledgel corresponds to more than one scene point, (4) fit a scene point to each linked edge list, (5) merge ledgels judged to correspond to the same point, and (6) compute a final scene point estimate for each merged ledgel. Even though there were some curved objects in the scene, we decided to keep the implementation simple by interpreting all ledgels as corresponding to stationary scene points.

Figure 53 (left) displays the original set of linked edge point lists. The edge points are the zero crossings of a difference-of-Gaussians operator; they are linked into ledgels on the basis of local information. The final estimates of the scene points after the processing of these ledgels are displayed in figure 53 (right). A simple least-squares fitting technique was used. Each scene point estimate is marked with an "x" and the scene model

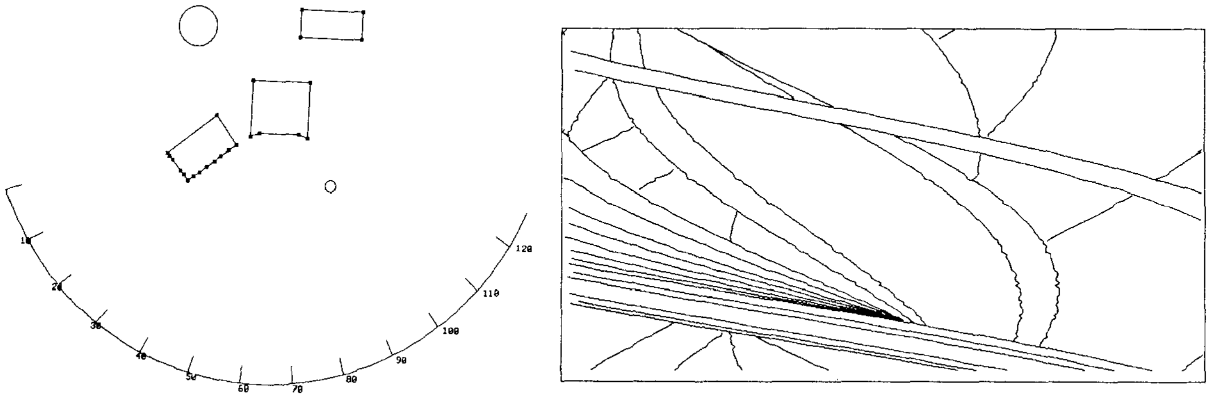


Fig. 52. Model of planar slice through scene and the robot's path (*left*); edges simulated by using model and path (*right*).

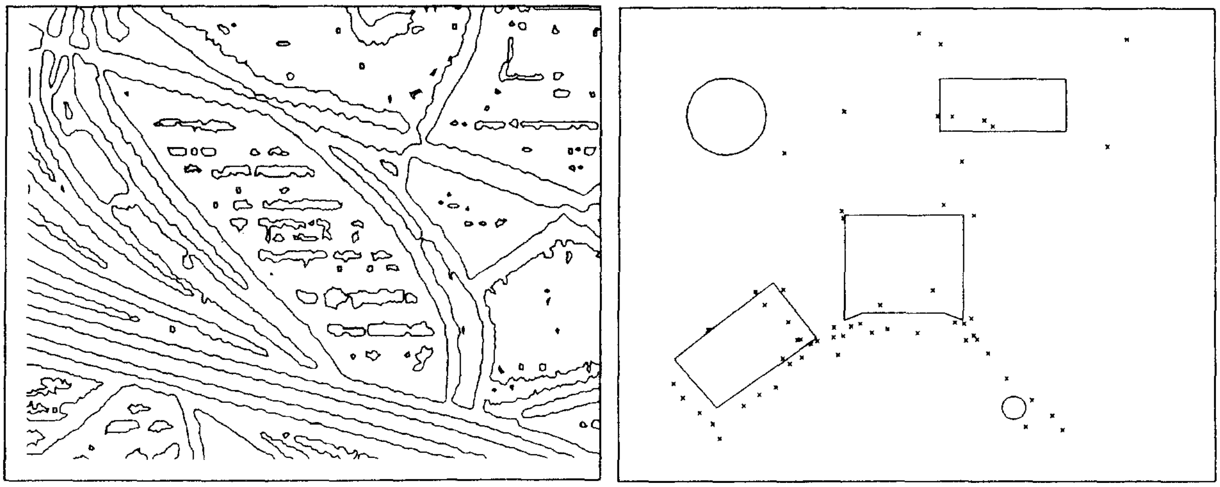


Fig. 53. Edges from the image sequence collected by the robot (*left*); estimated scene points (*right*).

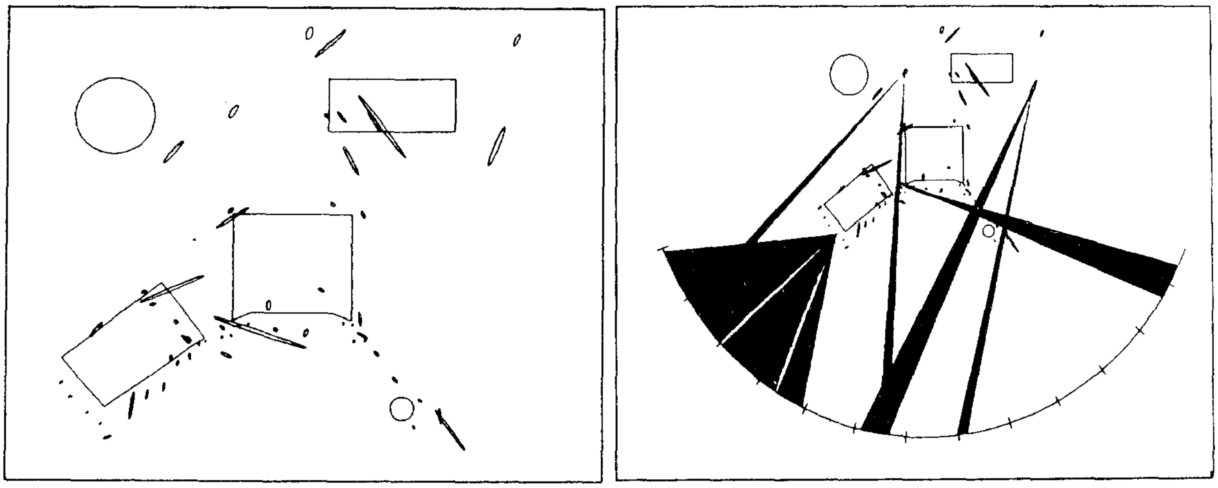


Fig. 54. Error ellipses of scene point estimates (*left*); wedges indicating ranges of camera locations from which some ellipses were viewed (*right*).

is superimposed for reference.

It is useful to have a measure of the uncertainty associated with each scene point estimate. The covariance of each estimate was computed as in the preceding section (see Marimont [57] for further details). Figure 54 (left) shows the covariances of the scene point estimates of figure 53 (right). As before, each covariance is displayed as an ellipse that should contain the true scene point at a confidence level exceeding 99%. Figure 54 (right) shows for a few of the scene points a wedge indicating from which camera locations the point at the vertex of the wedge was viewed. The size of the ellipses understates the uncertainty of the estimates because the ellipses account only for that portion of the uncertainty that is attributable to image noise, whereas in fact the camera path, the camera model (i.e., the focal length, optical distortion, area of the sensor, etc.), and even the model of the scene are all unreliable to varying degrees.

5.7 *Extensions to Space*

All the above results apply to one-dimensional views induced by a planar path through a planar scene. To generalize them to two-dimensional views induced by a three-dimensional path through a three-dimensional scene, we need the concept of duality in projective space. In this section, we briefly review duality in projective space and suggest some possible applications to the three-dimensional analogues of the problems studied above.

In projective space, a point in homogeneous coordinates (a, b, c, d) is dual to the plane that satisfies $ax_1 + bx_2 + cx_3 + dx_4 = 0$. The planes passing *through* a point $\mathbf{p}$ are dual to the points contained *in* the plane dual to $\mathbf{p}$. The dual of a line is another line; the planes that intersect in a line $\mathbf{l}$ are dual to points on the line dual to $\mathbf{l}$. The dual to a space curve consists of the duals to its osculating planes.

The tangent envelope of a differentiable surface consists of the planes tangent to the surface. The dual to a surface is the set of points dual to the planes of its tangent envelope. Quadric surfaces in space are analogous to conics in the plane; quadrics are second-order surfaces, while conics are second-order curves. Just as the dual of a conic is a conic, the dual of a quadric is a quadric.

We shall use a standard perspective-projection model of image formation in space: the location on the image plane of the image of a scene point is the intersection of the line through the camera center and the scene point with the image plane. For the moment, we consider images after edges have been detected, so that curves and their intersections are the only features in the image.

If the camera's position and orientation are known in some global coordinate system, so is the location of the image plane. The location of a feature in the image coordinate system on this plane thus implies the feature's location in the global coordinate system. From a point in the image, we can compute the parameters of the line of sight determined by the point and the camera center. From a line in the image, we can compute the parameters of the plane determined by the line and the camera center. Moreover, the line determined by a point in space and the camera center is the same as that determined by the projection of the point in space onto the image plane and the camera center, and the plane determined by a line in space and the camera center is the same as that determined by the projection of the line in space onto the image plane and the camera center. In the following, we shall refer to lines and planes determined by space points and lines and camera centers without mentioning that in fact they are computed by using image points and lines, which in general are the only data available.

We now apply duality to viewing the simplest geometric objects in space. Consider a point in space $\mathbf{p}$ viewed along a camera path that is a polygonal arc $\{\mathbf{q}_i\}$, as in figure 55. Let the line of sight through $\mathbf{p}$ and $\mathbf{q}_i$ be $\mathbf{l}_i$. Since all the $\mathbf{l}_i$ intersect at $\mathbf{p}$, their duals $\tilde{\mathbf{l}}_i$ lie in the plane $\tilde{\mathbf{p}}$ dual to $\mathbf{p}$. Because adjacent lines of sight $\mathbf{l}_i$ and $\mathbf{l}_{i+1}$ determine a plane $\mathbf{a}_i$, they intersect at the point $\tilde{\mathbf{a}}_i$ dual to $\mathbf{a}_i$. Since the next plane $\mathbf{a}_{i+1}$ is dual to $\tilde{\mathbf{a}}_{i+1}$, which is the intersection of $\mathbf{l}_{i+1}$ and $\mathbf{l}_{i+2}$, the $\tilde{\mathbf{a}}_i$ with $i = 1$, n form a polygonal arc. To estimate the location of a scene point from its views along a known camera path, we therefore estimate the plane in which lie the duals to the lines of sight.

Inferring the position of a line $\mathbf{l}$ by viewing it from along the [polygonal arc] camera path $\mathbf{q}_i$ is even simpler. Each camera location $\mathbf{q}_i$ and $\mathbf{l}$ determine a plane $\mathbf{a}_i$. Since these planes intersect in a line, their dual points $\tilde{\mathbf{a}}_i$ lie on the line $\tilde{\mathbf{l}}$ dual to $\mathbf{l}$. Thus, to estimate $\mathbf{l}$, we simply fit a line $\tilde{\mathbf{l}}$ to the points dual to the $\mathbf{a}_i$ and take its dual.

It is also possible to reconstruct a smooth surface by viewing its limbs from a smooth camera path. The planes determined by the camera center and the line tangent to each point on the limb are the subset of the surface's tangent envelope which is tangent to the limb. As the camera center moves, the limb sweeps out a section of the surface, and the tangent planes determined by the

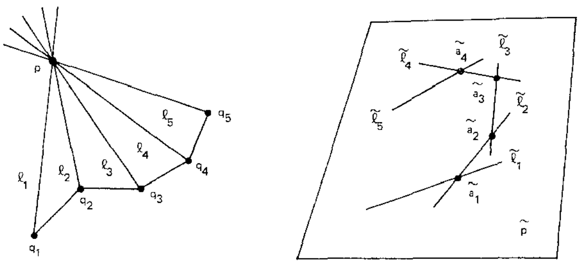


Fig. 55. Viewing a point along a camera path in space (*left*); duals of lines of sight (*right*).

tangents to the moving limb and the camera center sweep out a portion of the surface's tangent envelope. The points dual to the planes of that portion of the surface's tangent envelope also form a surface. The dual of the tangent envelope of this surface is the original surface; therefore, to recover the original surface, we fit a surface to the points dual to the original surface's tangent envelope, find its tangent envelope, and take its dual.

We have also obtained results for inferring the location (a) of a polygonal arc in space given a polygonal camera path and (b) a smooth curve in space given a smooth camera path (see Marimont [57] for details).

5.8 Discussion

In this section, we have sought to generalize the analysis of features in EPIs arising from linear camera motion to more general camera motion. There were a number of properties of EPI analysis in the linear case that we hoped to preserve. First, since each EPI corresponds to a planar slice of the scene, we can reconstruct the scene, one slice at a time, by analyzing each EPI independently. Second, there exists a simple linear relationship between features in the EPI and the location of the corresponding stationary scene points. Third, the topology of EPI edge features encodes the occlusions and disocclusions of scene objects.

Unfortunately, only linear camera motion preserves the stable epipolar planes that make scene reconstruction possible by analyzing EPIs independently. With planar camera motion, there is only one stable epipolar plane and one EPI, thus only one plane in the scene that can be reconstructed with these techniques. Here we have used projective duality in the plane to show that a "dual scene," consisting of the points dual to the lines of sight, is the appropriate extension of an EPI arising from linear camera motion. A simple linear relationship exists between features in the dual scene and the location of corresponding stationary scene points, and the topology of dual scene features encodes the occlusions and disocclusions of scene objects. Moreover, duality facilitated the analysis of objects other than stationary points, which were the only objects considered in the linear camera motion case. We have considered in detail both curved and independently moving objects.

Projective duality in space is necessary for the analysis of general camera paths in space; while the treatment here is necessarily superficial, we feel that the results from the planar motion case can be generalized in a reasonably straightforward way. The principal loss incurred in going from the handling of three-dimensional information in the linear motion case to that in the general motion case is that it is then no longer possible to decompose the reconstruction of the scene.

However satisfying these generalizations may be from a conceptual standpoint, their computational implications are unclear. By and large, an image feature and its dual are related nonlinearly, so that estimating scene features based on duals of noisy image features becomes problematic. An even more fundamental criticism is that duality is irrelevant to obtaining the results described in this section. In a sense, it is not really even a transformation, but simply a reinterpretation of the coordinates of geometric objects: points as lines, curves as tangent envelopes, and so on. (The nonlinearities are introduced when projective coordinates are interpreted as Euclidean coordinates.) If a problem is linear in the dual scene, it must have been linear in the original scene as well, so why not solve it there and thus avoid the costly and complicated nonlinear transformations?

In fact, the programs we have implemented compute directly with lines of sight and the scene features that give rise to them, rather than with their duals. But duality was critical to the analysis on which these programs are based; it facilitated the insights that suggested avenues for analysis, and it simplified details of the analyses themselves. Projective geometry is a rich and mature area of classical mathematics; establishing its relevance to the structure-from-motion problem is itself worthwhile, however small may be the direct effect on the programs we write.

Perhaps even more importantly, the use of duality in the case of planar camera motion led to a generalization of the analysis for a single EPI arising from linear camera motion. The generalization is complete in the sense that all aspects of the original EPI analysis are extended appropriately. Moreover, the generalization subsumes the original analysis because it is possible to show that, with the appropriate collineation, the dual of the lines of sight in an EPI arising from linear camera motion is precisely the EPI itself (see Marimont [57]). EPIs arising from linear camera motion are thus special cases of the general analysis: no transformation is necessary to take advantage of the inherent linearity in reconstructing the scene when the camera motion is known. In part, this is what makes these EPIs so attractive computationally: the linearity is explicit in the image, instead of being extractable solely through a nonlinear transformation (which depends exclusively on the camera motion).

6 Conclusion

EPI analysis is a simple, effective technique for building a three-dimensional description of a scene from a sequence of images. Its power is derived from two constraints on the imaging process. First, the camera is limited to a linear path (except for one special case of planar motion). Second, the image sequence is expected to contain a large number of closely spaced images. These two constraints make it possible to transform a difficult three-dimensional analysis into a set of straightforward two-dimensional analyses. The two-dimensional analyses involve only the detection of lines in images that contain approximately homogeneous regions bounded by lines.

EPI analysis combines spatial and temporal information in a fundamentally different way from most motion-analysis techniques. By taking hundreds of closely spaced images, we achieve a temporal continuity *between* images comparable to the spatial continuity *within* an image. This makes it possible to construct and analyze spatiotemporal images. Since these new images have essentially the same properties as conventional images, conventional techniques can be applied. For example, edge dectection techniques can be applied to EPIs, which are spatiotemporal images. However, the detection of an edge in one of these images is equivalent to selecting a feature in a spatial image and tracking it through several frames.

In addition to estimating the depths of scene features, EPI analysis provides two types of higher-level information about the scene. The first is a list of occlusion edges, the second a map of free space. Occlusion edges are important for segmenting the scene into solid objects; yet they have been difficult to identify with other image-understanding techniques. They are identified in EPI analysis by examining the patterns of line intersections in EPIs. The map of free space is an iconic representation of the scene that indicates which areas are known to be empty. It is constructed from regions swept out by lines of sight from the moving camera to scene features. These two types of information make it possible to build significantly more complete models of a scene than can be done from a mere list of three-dimensional points.

We believe that EPI analysis can be implemented efficiently for two reasons. First, the basic processing of each EPI is independent of other EPIs, making it possible to analyze them in parallel. Second, the analysis of an EPI involves only the detection of lines, which can be implemented efficiently in special-purpose hardware performing a Hough-type analysis. To make EPI analysis more practical, however, we plan to develop a version that works incrementally over time. This process, starting with an analysis similar to the one described in this article, would provide mechanisms for introducing new features as they appear and then updating the positions of previously detected features as they are tracked over time.

We are currently developing a technique for detecting and representing surfaces in the spatiotemporal block of data. These surfaces encode the spatial and temporal continuities along particular zero crossings in the data, thereby providing a direct way to link features from one EPI to another. This capability will make another crucial piece of information available for the modeling process. Furthermore, the surface detection operates incrementally over time. To explore the next step in this modeling process, we plan to develop techniques for representing three-dimensional objects that will enable the descriptions to gradually evolve as more detailed information is acquired.

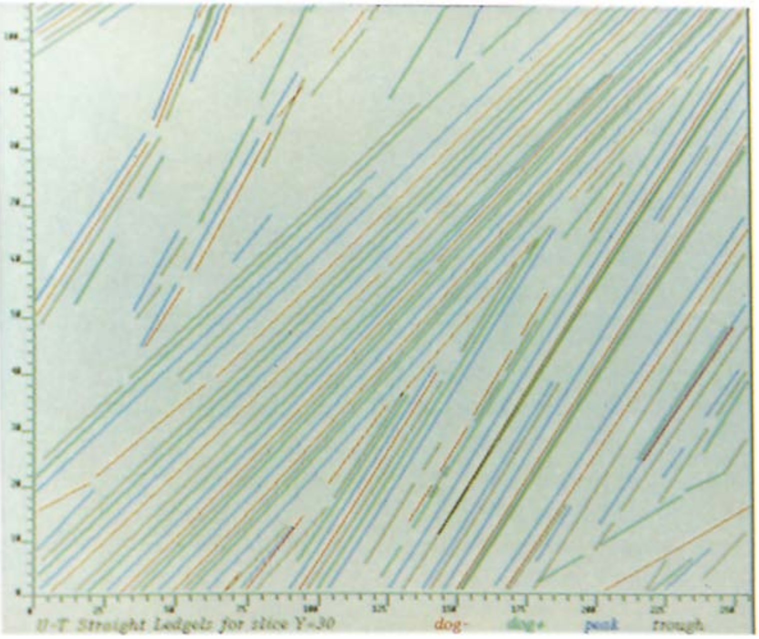


Fig. 56. Linear segments.

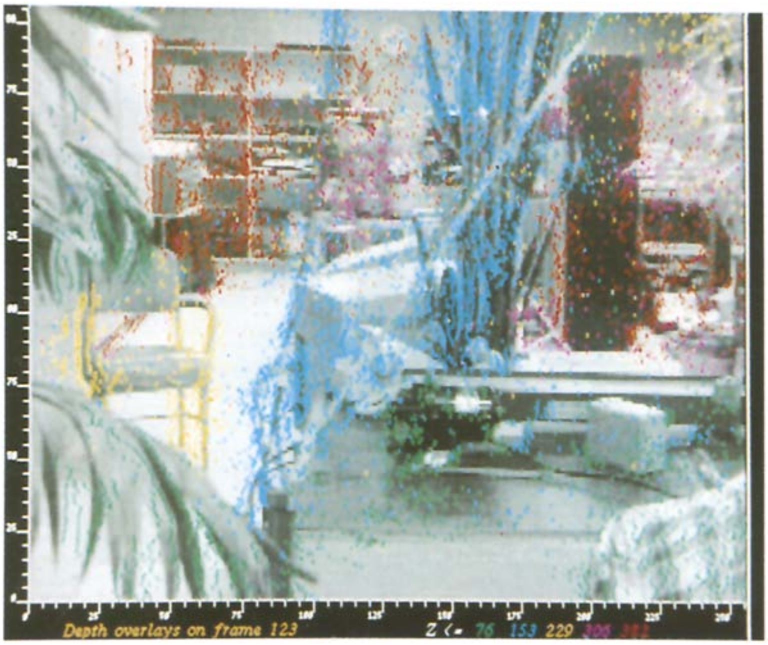


Fig. 57. Color-coded display of *z–y–z* points.

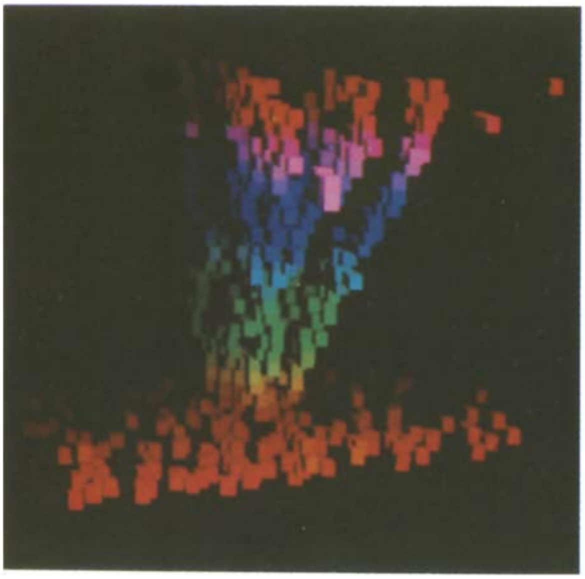


Fig. 58. Perspective image of voxels.

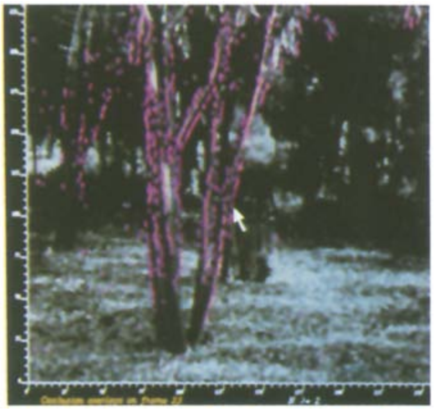


Fig. 59. Principal occlusion boundaries.

Bibliography

1. E.H. Adelson and J.R. Bergen, "Spatiotemporal energy models for the perception of motion," *Journal of the Optical Society of America A* **2**, pp. 284–299, 1985.
2. G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-7**, pp. 384–401, 1985.
3. J. Aloimonos and I. Rigoutsos, "Determining the 3-D motion of a rigid planar patch without correspondence, under perspective projection," in *Proceedings: Workshop on Motion: Representation and Analysis*, Kiawah Island, 1986, pp. 167–174.
4. T. Atherton, Report of private discussion with B. Steer of the University of Warwick, 1986.
5. L. Auslander and R.E. MacKenzie, *Introduction to Differentiable Manifolds*, New York: Dover, 1977.
6. H.H. Baker, R.C. Bolles, and D.H. Marimont, "A new technique for obtaining depth information from a moving sensor," in *Proceedings of the ISPRS Commission II Symposium on Photogrammetric and Remote Sensing Systems for Data Processing and Analysis*, Baltimore, 1986.
7. S.T. Barnard and W.B. Thompson, "Disparity analysis of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2**, 1980, pp. 333–340.
8. J. Barron, "A Survey of Approaches for Determining Optic Flow, Environmental Layout and Egomotion," Department of Computer Science, University of Toronto, Report No. RBCV-TR-84-5, 1984.
9. J.V. Beck and K.J. Arnold, *Parameter Estimation in Engineering and Science*, New York: John Wiley and Sons, 1977.
10. W.H. Beyer (ed.), *CRC Standard Mathematical Tables*, 26th edn., Boca Raton, FL: CRC Press, 1981.
11. A.P. Blicher and S.M. Omohundro, "Unique recovery of motion and optic flow via lie algebras," in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, Los Angeles, 1985, pp. 889–891.
12. R.C. Bolles, and H.H. Baker, "Epipolar-plane image analysis: A technique for analyzing motion sequences," in *Proceedings of the Third Workshop on Computer Vision: Representation and Control*, Bellaire, MI, 1985, pp. 168–178.
13. N.J. Bridwell and T.S. Huang, "A discrete spatial representation for lateral motion stereo," *Computer Vision, Graphics, and Image Processing* **21**, pp. 33–57, 1983.
14. T.J. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**, pp. 90–99, 1986.
15. T.J. Broida and R. Chellappa, "Kinematics and structure of a rigid object from a sequence of noisy images," in *Proceedings: Workshop on Motion: Representation and Analysis*, Kiawah Island, 1986, pp. 95–100.
16. A.R. Bruss and B.K.P. Horn, "Passive navigation," *Computer Vision, Graphics, and Image Processing* **21**, pp. 3–20, 1983.
17. B.F. Buxton and H. Buxton, "Monocular depth perception from optical flow by space time signal processing," *Proceedings of the Royal Society of London B* **218**, pp. 27–47, 1983.
18. B.F. Buxton and H. Buxton, "Computation of optic flow from the motion of edge features in image sequences," *Image and Vision Computing* **2**, 1984, pp. 59–75.
19. R. Chatila and J.-P. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, St. Louis, 1985, pp. 138–145.
20. H.S.M. Coxeter, *Projective Geometry*, New York: Blaisdell, 1964.
21. L.S. Dreschler and H.-H. Nagel, "Volumetric model and 3-D trajectory of a moving car derived from monocular TV-frame sequences of a street scene," in *Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, Vancouver, 1981, pp. 692–697.
22. L.S. Dreschler and H.-H. Nagel, "On the selection of critical points and local curvature extrema of region boundaries for interframe matching," in *Image sequence Processing and Dynamic Scene Analysis*, T.S. Huang (ed.), Berlin: Spring-Verlag, 1983, pp. 457–470.
23. J.-Q. Fang and T.S. Huang, "Solving three-dimensional small-rotation motion equations: Uniqueness, algorithms, and numerical results, *Computer Vision, Graphics, and Image Processing* **26**, pp. 183–206, 1984.
24. J.-Q. Fang and T.S. Huang, "Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**, pp. 545–554, 1984.
25. C.L. Fennema and W.B. Thompson, "Velocity determination in scenes containing several moving objects," *Computer Graphics and Image Processing* **9**, pp. 301–315, 1979.
26. D.B. Gennery, "Tracking known three-dimensional objects," in *Proceedings of the Second National Conference on Artificial Intelligence (AAAI-82)*, Pittsburgh, 1982, pp. 13–17.
27. M.J. Hannah, "Bootstrap stereo," in *Proceedings of the Image Understanding Workshop*, College Park, MD, 1980, pp. 201–208.
28. S.M. Haynes and R. Jain, "Detection of moving edges," *Computer Vision, Graphics, and Image Processing* **21**, 1983, pp. 345–367.
29. S.M. Haynes and R. Jain, "Low level motion events: trajectory discontinuities," in *Proceedings of the First Conference on AI Applications*, Denver, 1984, pp. 251–256.
30. D.J. Heeger, "Depth and flow from motion energy," in *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, Philadelphia, 1986, pp. 657–663.
31. E.C. Hildreth, "Computations underlying the measurement of visual motion," *Artificial Intelligence* **23**, pp. 309–354, 1984.
32. E.C. Hildreth and N.M. Grzywacz, "The incremental recovery of structure from motion: Position vs. velocity based formulations," in *Proceedings: Workshop on*

Motion: Representation and Analysis, Kiawah Island, 1986, pp. 137–143.

33. D.D. Hoffman, "Inferring local surface orientation from motion fields," *Journal of the Optical Society of America* **72**, pp. 888–892, 1982.
34. D.D. Hoffman and B.E. Flinchbaugh, "The interpretation of biological motion," *Biological Cybernetics* **42**, pp. 195–204, 1982.
35. B.K.P. Horn and B.G. Schunk, "Determining optical flow," *Artificial Intelligence* **17**, pp. 185–203, 1981.
36. B.K.P. Horn and E.J. Weldon, Jr., "Robust direct methods for recovering motion," unpublished manuscript, February 1986.
37. T.S. Huang (ed.), *Image sequence Analysis*, Berlin: Springer-Verlag, 1981.
38. D.A. Huffman, "Impossible objects as nonsense sentences," *Machine Intelligence* **6**, pp. 295–324, 1971.
39. D.A. Huffman, "A duality concept for the analysis of polyhedral scenes," *Machine Intelligence* **8**, pp. 475–492, 1977.
40. R.Jain, "Direct computation of the focus of expansion," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-5**, pp. 58–63, 1983.
41. R. Jain, "Detection on moving edges" in *Proceedings of the First Conference on AI Applications*, Denver, 1984, pp. 142–148.
42. K. Kanatani, "Tracing planar surface motion from projection without knowing correspondence," *Computer Vision, Graphics, and Image Processing* **29**, pp. 1–12, 1985.
43. K. Kanatani, "Detecting the motion of a planar surface by line and surface integrals," *Computer Vision, Graphics, and Image Processing* **29**, pp. 13–22, 1985.
44. K. Kanatani, "Structure from motion without correspondence: General principle," in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, Los Angeles, 1985, pp. 886–888.
45. K. Kanatani, "Transformation of optical flow by camera rotation," in *Proceedings: Workshop on Motion: Representation and Analysis*, Kiawah Island, 1986, pp. 113–118.
46. J.J. Koenderink and A.J. van Doorn, "Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer," *Optica Acta* **22**, pp. 773–791, 1975.
47. J.J. Koenderink and A.J. van Doorn, "Local structure of movement parallax of the plane," *Journal of the Optical Society of America* **66**, pp. 717–723, 1976.
48. J.J. Koenderink and A.J. van Doorn, "The singularities of the visual mapping," *Biological Cybernetics* **24**, pp. 51–59, 1976.
49. J.J. Koenderink and A.J. van Doorn, "How an ambulant observer can construct a model of the environment from the geometrical structure of the visual inflow," in *Kybernetik 77*, Hauske and Butenandt (eds.), 1977, pp. 224–227.
50. J.J. Koenderink and A.J. van Doorn, "Exterospecific component of the motion parallax field," *Journal of the Optical Society of America* **71**, pp. 953–957, 1981.
51. D.T. Lawton, "Processing translational motion sequences," *Computer Vision, Graphics, and Image Processing* **22**, pp. 116–144, 1983.
52. Y. Liu and T.S. Huang, "Estimation of rigid body motion using straight line correspondences," in *Proceedings: Workshop on Motion: Representation and Analysis*, Kiawah Island, 1986, pp. 47–52.
53. H.D. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature* **293**, pp. 133–135, 1981.
54. H.D. Longuet-Higgins, "The visual ambiguity of a moving plane," *Proceedings of the Royal Society of London B* **223**, pp. 165–175, 1984.
55. H.D. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proceedings of the Royal Society of London B* **208**, pp. 385–397, 1980.
56. A.K. Mackworth, "Interpreting pictures of polyhedral scenes," *Artificial Intelligence* **4**, pp. 121–137, 1977.
57. D.H. Marimont, "Inferring spatial structure from feature correspondences," PhD dissertation, Stanford University, 1986.
58. D.H. Marimont, "Projective duality and the analysis of image sequences," in *Proceedings: Workshop on Motion: Representation and Analysis*, Kiawah Island, 1986, pp. 7–14.
59. D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London B* **207**, pp. 187–217, 1980.
60. E.A. Maxwell, *The Methods of Plane Projective Geometry Based on the Use of General Homogeneous Coordinates*, Cambridge: Cambridge University Press, 1946.
61. E.A. Maxwell, *General Homogeneous Coordinates in Space of Three Dimensions*, Cambridge: Cambridge University Press, 1959.
62. J.L. Melsa and D.L. Cohn, *Decision and Estimation Theory*, New York: McGraw-Hill, 1978.
63. A. Mitiche, S. Seida, and J.K. Aggarwal, "Line-based computation of structure and motion using angular invariance," in *Proceedings: Workshop on Motion: Representation and Analysis*, Kiawah Island, 1986, pp. 175–180.
64. H.P. Moravec, "Visual mapping by a robot rover," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Tokyo, 1979, pp. 598–600.
65. H.P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, St. Louis, 1985, pp. 116–120.
66. H.-H. Nagel, "Displacement vectors derived from second-order intensity variations in image sequences," *Computer Vision, Graphics, and Image Processing* **21**, pp. 85–117, 1983.
67. S. Negahdaripour and B.K.P. Horn, "Direct passive navigation," MIT Artif. Intell. Lab., Massachusetts Inst. Technol., AI Memo 821, February 1985.
68. S. Negahdaripour, "Direct passive navigation: analytical solution for planes," MIT Artif. Intell. Lab., Massachusetts Inst. Technol., AI Memo 863, August 1985.
69. R. Nevatia, "Depth measurement from motion stereo," *Computer Graphics and Image Processing* **5**, 1976, pp.

203–214.

70. A.P. Pentland, "Perceptual organization and the representation of natural form," *Artificial Intelligence* **28**, pp. 293–331, 1986.
71. K. Prazdny, "Motion and structure from optical flow," in *Proceedings of the Sixth International Joint Conference on Artificial Intelligence (IJCAI-79)*, Tokyo, 1979, pp. 702–704.
72. K. Prazdny, "Egomotion and relative depth map from optical flow," *Biological Cybernetics* **36**, pp. 87–102, 1980.
73. K. Prazdny, "Determining the instantaneous direction of motion from optical flow generated by a curvilinearly moving observer," *Computer Graphics and Image Processing* **17**, pp . 238–248, 1981.
74. K. Prazdny, "On the information in optical flows," *Computer Vision, Graphics, and Image Processing* **22**, pp. 239–259, 1983.
75. K. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer Graphics and Image Processing* **1**, 1972, pp. 224–256.
76. J.W. Roach and J.K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-2**, pp. 554–562, 1980.
77. J.W. Roach and J.S. Wright, "Spherical dual images: A 3D representation method for solid objects that combines dual space and Gaussian spheres," in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, 1986, pp. 1087–1092.
78. M. Subbarao, "Interpretation of image motion fields: A spatio-temporal approach," in *Proceedings: Workshop on Motion: Representation and Analysis*, Kiawah Island, 1986, pp. 157–165.
79. M. Subbarao and A.M. Waxman, "On the uniqueness of image flow solutions for planar surfaces in motion," in *Proceedings of the Third Workshop on Computer Vision: Representation and Control*, Bellaire, MI, 1985, pp. 129–140.
80. E.H. Thompson, "A rational algebraic formulation of the problem of relative orientation," *Photogrammetric Record* **3**, pp. 152–159, 1959.
81. W.B. Thompson and S.T. Barnard, "Lower-level estimation and interpretation of visual motion," *Computer* **14**, 1981, pp. 20–28.
82. W.B. Thompson, K.M. Mutch, and V.A. Berzins, "Dynamic occlusion analysis in optical flow fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-7**, pp. 374–383, 1985.
83. R.Y. Tsai, "Estimating 3-D motion parameters and object surface structures from the image motion of conic arcs. I: Theoretical basis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Boston, 1983, pp. 122–125.
84. R.Y. Tsai, "Estimating 3-D motion parameters and object surface structures from the image motion of curved edges," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, 1983, pp. 259–266.
85. R.Y. Tsai and T.S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE Transactions on Acoustics, Speech, and Signal Processing* **29**, pp. 1147–1152, 1981.
86. R.Y. Tsai and T.S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch. II: Singular value decomposition," *IEEE Transactions on Acoustics, Speech, and Signal Processing* **30**, pp. 525–533, 1982.
87. 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**, pp. 13–27, 1984.
88. S. Ullman, *The Interpretation of Visual Motion*, Cambridge, MA: MIT Press, 1979.
89. S. Ullman, "Recent computational studies in the interpretation of structure from motion," in *Human and Machine Vision*, J. Beck, B. Hope, and A. Rosenfeld (eds.), Orlando, FL: Academic Press, 1983, pp. 459–480.
90. S. Ullman, "Maximizing rigidity: The incremental recovery of 3-D structure from rigid and rubbery motion," *Perception* **13**, pp. 255–274, 1984.
91. A.M. Waxman, "An image flow paradigm," in *Proceedings of the Second IEEE Workshop on Computer Vision: Representation and Control*, Annapolis, 1984, pp. 49–57.
92. A.M. Waxman, and S. Ullman, "Surface structure and 3-D motion from image flow: A kinematic analysis," Center for Automation Research, University of Maryland, CAR Tech. Report 24, October 1983.
93. A.M. Waxman, and S. Ullman, "Surface structure and three-dimensional motion from image flow kinematics," *International Journal of Robotics Research* **4**, pp. 72–94, 1985.
94. A.M. Waxman and K. Wohn, "Contour evolution, neighborhood deformation, and global image flow: Planar surfaces in motion," Center for Automation Research, University of Maryland, CAR Tech. Report 58, April 1984.
95. A.M. Waxman and K. Wohn, "Contour evolution, neighborhood deformation, and global image flow: Planar surfaces in motion," *International Journal of Robotics Research* **4**, pp. 95–108, 1985.
96. J.A. Webb and J.K. Aggarwal, "Visually interpreting the motion of objects in space," *Computer* **14**, pp. 40–46, 1981.
97. K.W. Wong (author–editor), "Basic mathematics of photogrammetry," in *Manual of Photogrammetry*, 4th edn., Falls Church, VA: American Society of Photogrammetry, 1980.
98. C.R. Wylie, Jr., *Introduction to Projective Geometry*, New York: McGraw-Hill, 1970.
99. M. Yamamoto, "Motion analysis using the visualized locus method," untranslated Japanese articles, 1981.
100. B.L. Yen and T.S. Huang, "Determining 3-D motion and structure of a rigid body using the spherical projec-

tion," *Computer Graphics and Image Processing* **21**, pp. 21–32, 1983.

01. B.L. Yen and T.S. Huang, "Determining the 3-D motion and structure of a rigid body using straight line correspondences," *Image sequence Processing and Dynamic Scene Analysis*, T.S. Huang (ed.), Berlin: Spring-Verlag, pp. 365–394, 1983.