SRI International

ONE-EYED STEREO: A UNIFIED STRATEGY TO RECOVER SHAPE FROM A SINGLE IMAGE

Technical Note No. 367

5 November 1985

By: Thomas M. Strat, Computer Scientist
Martin A. Fischler, Program Director, Perception

Artificial Intelligence Center
Computer Science and Technology Division

SRI Project 5355

The work reported herein was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. MDA903-83-C-0027.

SRI International®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Abstract

A single two-dimensional image is an ambiguous representation of the three-dimensional world—many different scenes could have produced the same image—yet the human visual system is extremely successful at recovering a qualitatively correct depth model from this type of representation. Workers in the field of computational vision have devised a number of distinct schemes that attempt to emulate this human capability; these schemes are collectively known as "shape from ...." methods (*e.g.*, shape from shading, shape from texture, or shape from contour). In this paper we contend that the distinct assumptions made in each of these schemes must be tantamount to providing a second (virtual) image of the original scene, and that any one of these approaches can be translated into a conventional stereo formalism. In particular, we show that it is frequently possible to structure the problem as one of recovering depth from a stereo pair consisting of the supplied perspective image (the *original* image) and an hypothesized orthographic image (the *virtual* image). We present a new algorithm of the form required to accomplish this type of stereo reconstruction task.

1 Introduction

The recovery of 3-D scene geometry from one or more images, which we will call the scene-modeling problem (SMP), has solutions that appear to follow one of three distinct paradigms: stereo; optic flow; and shape from shading, texture, and contour.

In the stereo paradigm, we match corresponding world/scene points in two images and, given the relative geometry of the two cameras (eyes) that acquired the images, we can use simple trigonometry to determine the depths of the matched points [1].

In the optic-flow paradigm, we use two or more images to compute the image velocity of corresponding scene points. If the camera's motion and imaging parameters are known, we can again use simple trigonometry to convert velocity measurements in the image to depths in the scene [21].

In the shape from shading, texture, and contour (SSTC) paradigm, we must either know, or make some assumptions about the nature of the scene, the illumination, and the imaging geometry. Brady's 1981 volume on computer vision [2] contains an excellent collection of papers, many of which address the problem of how to recover depth from the shading, texture, and contour information visible in a single image. Two distinct computational approaches have been employed in the SSTC paradigm: (1) integration of partial differential equations describing the relation of shading in an image to surface geometry in a scene, and (2) back-projection of planar image facets to undo the distortion in an image attribute (*e.g.*, edge orientation) induced by the imaging process on an assumed scene property (*e.g.*, uniform distribution of edge orientations).

Our purpose in this paper is to provide a unifying framework for the scene modeling problem, and to present a new computational approach to recovering scene geometry from the shading, texture, and contour information in a single image. Our contribution is based on the following observation: regardless of the assumptions employed in the SSTC paradigm, if a 3-D scene model has been derived successfully, it will generally be possible to establish a large number of correspondences between image and scene (model) points. From these correspondences we can compute a collineation matrix [11], and then extract the imaging geometry from it [4] [19]. We can

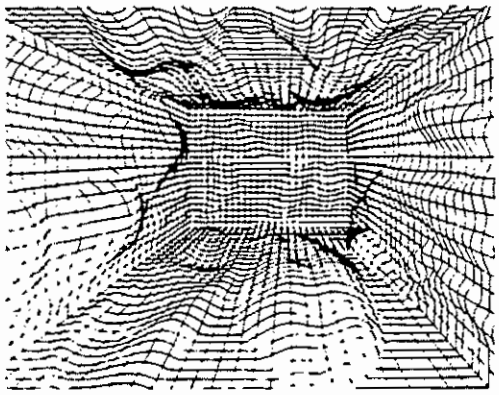

Figure 1: Wire Room

now construct a second image of the scene as viewed by the camera from some arbitrary location in space. It is thus obvious that any technique that is competent to solve the SMP must either be provided with at least two images or make assumptions that are equivalent to providing a second image. We can unify the various approaches to the SMP by converting their respective assumptions and auxiliary information into the implied second image and employing the stereo paradigm to recover depth. In the case of the SSTC paradigm, our approach amounts to "one-eyed stereo."

2 Shape from One-Eyed Stereo

Most people viewing Figure 1 get a strong impression of depth. We can recover an equivalent depth model by assuming that we are viewing a projection of a uniform grid and employing the computational procedure to be described. In the remainder of this paper we will show how some simple modifications and variations of the uniform grid, as the implied second image, allow us to recover depth from shading, texture, and contour.

The one-eyed stereo paradigm can be described as a five-step process, as outlined in the paragraphs below. Some scenes with special surface markings or image-formation processes must be analyzed by variants of the algorithm described, but the general approach remains the same.

2.1 Partition the Image

As with all approaches to the SMP, the image must be segmented into regions prior to the application of a particular algorithm. Before the one-eyed stereo computation can be employed, the segmentation process must delineate regions that are individually in conformance with a single model of image formation. The computation can then be carried out independently in each region, and the results fitted together.

2.2 Select a Model

For each region identified by the partitioning process, we must decide upon the underlying model of image formation that explains that portion of the image. Surface reflectance functions and texture patterns are examples of such models. Partitioning of the image and selection of the appropriate models are difficult tasks that are not addressed in this paper. Witkin and Kass [23] are exploring a new class of techniques that promises some eventual answers to these questions. Generally, it will be impossible to recover depth whenever a single model cannot be associated with a region. Similarly, inaccurate or incorrect results can be expected if the partitioning or modeling is performed incorrectly.

2.3 Generate the Virtual Image

The key to one-eyed stereo is using the model of image formation to fabricate a second (virtual) image of the scene. The idea is that the model often allows one to construct an image that is independent of the actual shape of the imaged surface. This allows the virtual image to be depicted solely from knowledge of the model without making use of the original image. For example, the markings on the surface of Figure 2(a) could have arisen from projection of a uniform grid upon the surface. For all images that fit this model, we can use a uniform grid as the virtual image. As a rule, the orientation, position, and scale of this grid will be unknown; however, we will show how this information can be recovered from the original image. Other models give rise to other forms of virtual images.

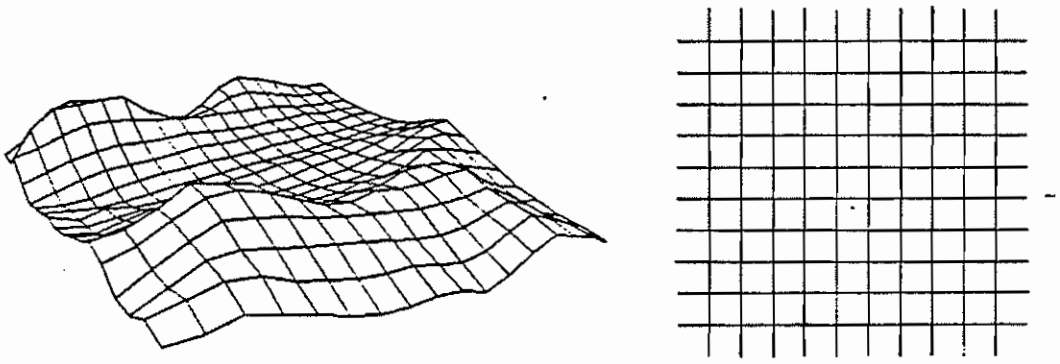

Figure 2: (a) A projected texture (b) Its virtual image

2.4 Determine Correspondences

Before applying stereo techniques to calculate depths, we must first establish correspondences between points in the real image and the virtual image. When dealing with textures, the process is typified by counting texels in each image from a chosen starting point. With shaded images, the general approach is to integrate intensities. Several variants of the method for establishing correspondences are described in the next section. The difficulty of the procedure, it should be noted, will depend on the nature of the model.

2.5 Compute Depths Using Stereo

With two images and a number of point-to-point correspondences in hand, the techniques of binocular stereo are immediately applicable. At this point, the problem has been reduced to computing the relative camera models between the two images and using that information to compute depths by triangulation. The fact that the virtual image will normally be an orthographic projection required reformulation of existing algorithms for performing this computation. The appendix describes a new algorithm that computes the relative camera model and reconstructs the 3-D scene from eight point correspondences between a perspective and an orthographic image.

The problem of recovering scene and imaging geometry from two or

more images has been addressed by workers not only in binocular stereo, but also in monocular perception of motion in which the two projections are separated in time as well as space. Various approaches have been employed to derive equations for the 3-D coordinates and motion parameters; these equations are generally solved by iterative techniques [5] [8] [13] [14]. Ullman [21] presents a solution for recovering 3-D shape from three orthographic projections with established correspondences among at least four points. His "polar equation" allows computation of shape when the motion of the scene is restricted to rotation about the vertical axis with arbitrary translation. Nagel and Neumann [10] have devised a compact system of three nonlinear equations for the unrestricted problem when five point correspondences between the two perspective images are known. More recently, Huang [20] and Longuet-Higgins [9] have independently derived methods requiring only that a set of eight simultaneous linear equations be solved when eight point correspondences between two perspective images are known. In our formulation we are faced with a stereo problem involving a perspective and an orthographic image; while the aforementioned references are indeed germane, none provides a solution to this particular problem.

The derivation described in the appendix was inspired by the formulation of Longuet-Higgins for perspective images. When either image nears orthography, Longuet-Higgins's method becomes unstable; it is undefined if either image is truly orthographic. Moreover, his approach requires knowledge of the focal length and principal point in each image while our method was derived specifically for one orthographic and one perspective image whose internal imaging parameters may not be fully known.

3 Variations on the Theme

In this section we illustrate how our approach is used with several models of texture, shading, and contour. Where these models do not match given scene characteristics, they may require additional modification. However, a qualitatively correct answer might still be obtainable by applying one of the specific models we discuss below to a situation that appears to be inappropriate, or to an image in which the validity of the assumptions

cannot be established.

3.1 Shape from Texture

Surface shapes are often communicated to humans graphically by drawings like Figure 2(a). Such illustrations can also be interpreted by one-eyed stereo. In this case, there is no need to partition the image; the underlying model of the entire scene consists of the intersections of lines distributed in the form of a square grid. When viewed directly from above at an infinite distance, the surface would appear as shown in the virtual image of Figure 2(b) regardless of the shape of the surface. This virtual image can be construed as an orthographic projection of the object surface from a particular, but unknown, viewing direction. Correspondences between the original and virtual images are easily established if there are no occlusions in the original image. Select any intersection in the original image to be the reference point and pair it with any intersection in the virtual image. A second corresponding pair can be found by moving to an adjacent intersection in both images. Additional pairs are found in the same manner, being careful to correlate the motions in each image consistently in both directions. When occlusions are present, it may still be possible to obtain correspondences for all visible junctions by following a nonoccluded path around the occlusion (such as the hill in the foreground of Figure 2(a)). If no such path can be found, the shape of each isolated region can still be computed, but there will be no way to relate the distances without further information. Other techniques used to represent images of 3-D shapes graphically may require other virtual images. Figure 3(a), for example, would imply a virtual image as shown in Figure 3(b). Methods for recognizing which model to apply are needed, but are not discussed here.

Once correspondences have been determined, we can use the algorithm given in the appendix to recover depth. We have presumably one perspective image and one orthographic image whose scale and origin are still unknown. The depths to be recovered will be scaled according to the scale chosen for the virtual image[1]. The choice of origin for the orthographic im-

[1] Recall that the original image does not contain the information necessary to recover the absolute size of the scene.

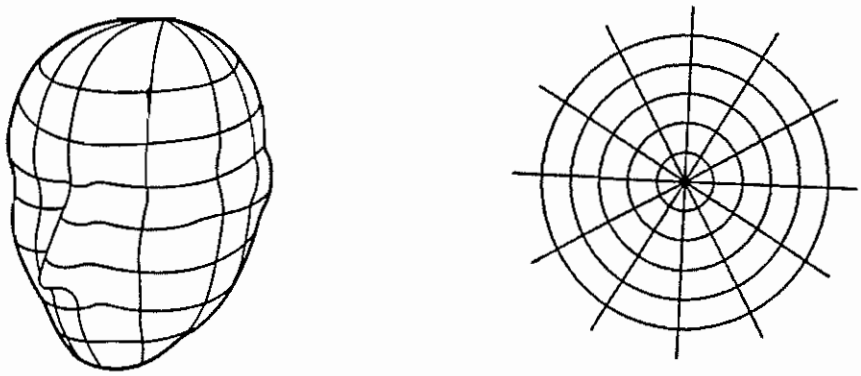

Figure 3: (a) The original image (b) The virtual image

age is arbitrary, and will lead to the same solution regardless of the point chosen. The appendix shows how to compute both the orientation and the displacement of the orthographic coordinate system, relative to the perspective imaging system. 3-D coordinates of each matched point are then easily computed by means of back-projection. A unique solution will be obtained whenever the piercing point or focal length of the perspective image is known. A minimum of eight pairs of matched points is required to obtain a solution; depths can be computed for all matched points.

There exists a growing literature on methods to recover shape from natural textures [7][12][18][22]. We will now show how the constraints imposed by one type of natural texture can be exploited to obtain similar results by using one-eyed stereo.

Consider the pattern of streets in Figure 4. If this city were viewed from an airplane directly overhead at high altitude, the streets would form a regular grid not unlike the one used as the virtual image in Figure 2. There are many other scene attributes that satisfy this same model. The houses in Figure 5 would appear to be distributed in a uniform grid if viewed from directly overhead. In an apple orchard growing on a hillside, the trees would be planted in rows that are evenly spaced when measured horizontally; the vineyard in Figure 6 exhibits this property.

Ignoring the nontrivial tasks of partitioning these images into isotextural regions, verifying that they satisfy the model, and identifying individual texels, it can be seen how these images can be interpreted with the same

Figure 4: The streets in this scene resemble a projected texture. [3]

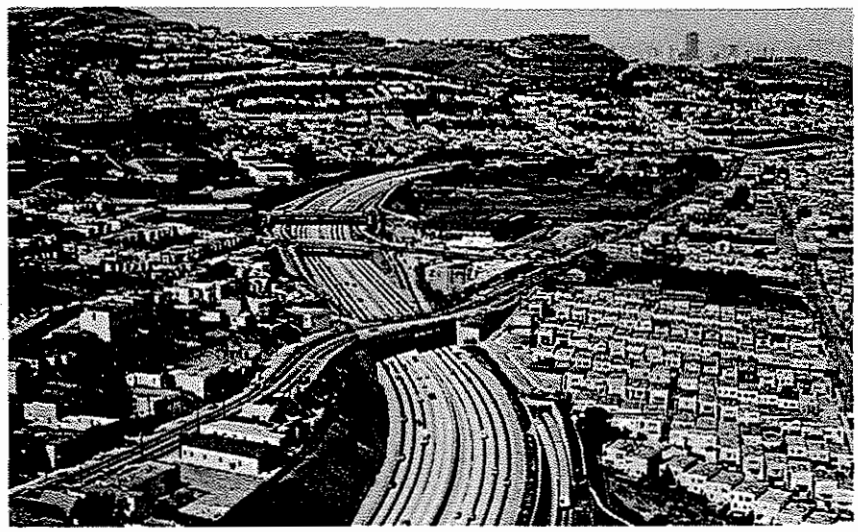

Figure 5: The houses can be construed as a projected texture. [3]

Figure 6: These grapevines exhibit a regular texture. [3]

techniques as were described in the previous section. The virtual image in each case will be a rectangular grid that can be considered as an orthographic view from an unknown orientation. Correspondences can be established by counting street intersections, rooftops, or grape vines. As before, one can solve for the relative camera model and compute depths of matched points. Obviously, for the situations discussed here, we must be satisfied with a qualitatively correct interpretation—not only because of the difficulty of locating individual texels reliably and accurately, but also in view of the numerical instabilities arising from the underlying nonlinear transformation.

3.2 Shape from Shading

For our purposes, surface shading can be considered the limiting case of a locally uniform texture distribution (as the texels approach infinitesimal dimensions). To compute correspondences, we need to integrate image intensities appropriately in place of counting lines, since the image intensities can be seen to be related to the density of lines projected on the surface. The feasibility of this procedure depends on the reflectance function of the surface.

What types of material possess the special property that allows their images to be treated like the limiting case of the projected textures of the previous section? The integral of intensity in an image region has to be proportional to the number of texels that would be projected in that region. If the angles i and e are defined as depicted in Figure 7, it can be seen that the number of texels projected onto a surface patch will be proportional to $\cos i$, the cosine of the incident angle. At the same time, the surface patch (as seen from the viewpoint) will be foreshortened by $\cos e$, the cosine of the emittance angle. Thus, the integral of reflected light intensity over a region will be proportional to the flux of the light striking the surface if the intensity of the reflected light at any point is proportional to $\cos i / \cos e$. Horn [6] has pointed out that, when viewed from great distances, the material in the maria of the moon and other rocky, dusty objects exhibit a reflectance function that allows recovery of the ratio $\cos i / \cos e$ from the imaged intensities. This surface property

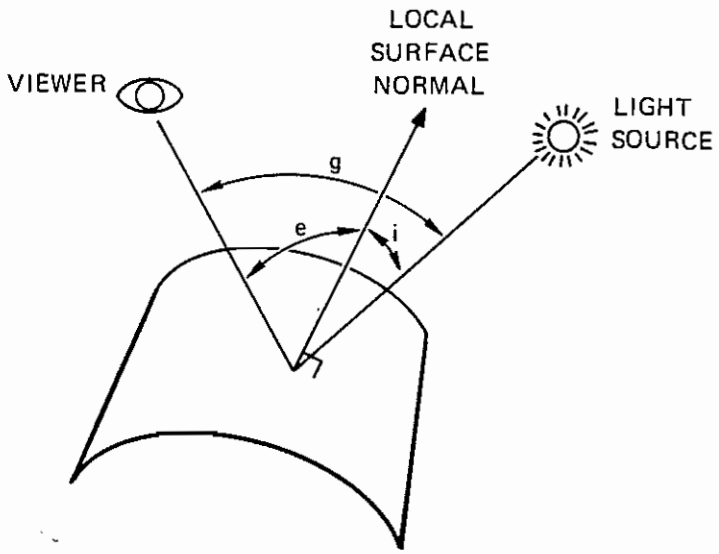

Figure 7: The geometry of surface illumination

has made possible unusually simple algorithms for computing shape-from-shading, so it is not surprising that it submits easily to one-eyed stereo as well.

To interpret this type of shading, we can construct a virtual image whose direction of view is the lighting direction (*i.e.*, taken from a "virtual camera" located at the light source). When the original shaded image is orthographic, we consider a family of parallel lines in which each line lies in a plane that includes both the light source and the (distant) viewpoint. When viewed from the light source, the image of the surface corresponding to these lines will also be a set of parallel lines regardless of the shape of the surface. These parallel lines constitute the virtual image. We will use the image intensities to refine these line-to-line correspondences to point-to-point correspondences. Figure 8 shows the geometry for an individual line in the family. A little trigonometry shows that

$$\Delta s' = \frac{\cos i}{\cos e} \Delta s \qquad , \tag{1}$$

where

Δs is a distance along the line in the real image and $\Delta s'$ is the corre-

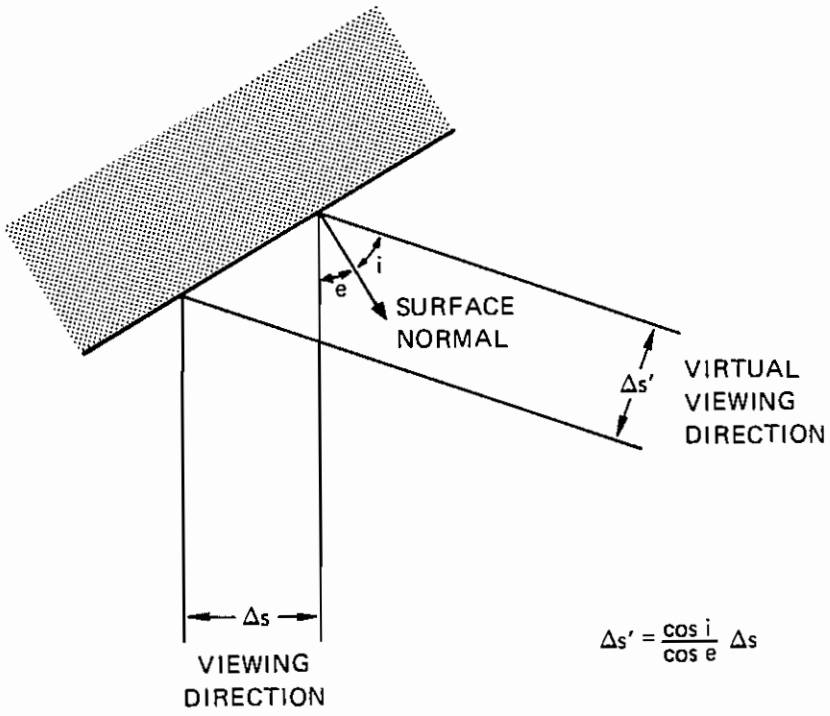

Figure 8: The geometry along a line in the direction of the light source

sponding distance along the corresponding line in the virtual image. Integrating this equation produces the following expression, which defines the point correspondences in the two images along the given line.

$$s' = s'_o + \int_0^s \frac{\cos i}{\cos e} ds \tag{2}$$

To use this equation we must first compute $\frac{\cos i}{\cos e}$ from the intensity value at each point along the line. This will, of course, be possible only when the reflectance function is constant for constant $\frac{\cos i}{\cos e}$. Next we choose a starting point in the shaded image and begin integrating intensities according to Equation (2). For any value of s, the corresponding virtual image point is along a straight line at a distance s' from the virtual reference point. With these point-to-point correspondences in hand, it is a simple matter of triangulation to find the 3-D coordinates of the surface points, given that we know the direction to the light source. We can explore the remainder of

the surface by repeating the process for each of the successive parallel lines in the image. Adjacent profiles still remain unrelated to each other, since their individual scale factors have not yet been ascertained. Knowledge of the actual depth of one point along each profile provides the necessary additional information to complete the reconstruction. It is important to note that our assumptions and initial conditions are those used by Horn; the fact that he was able to obtain a solution under these conditions assured the existence of a suitable virtual image for the one-eyed stereo paradigm.

For shaded perspective images, we must integrate along a family of straight lines that radiate from the point in the image that corresponds to the location of the light source. This ensures that the image line will be in a plane containing both the viewer and the light source, and that the virtual image of each line will also be a straight line. The integration becomes a bit more complex than shown in Equation 2 because the nonlinear effects of perspective imaging must be accommodated. Nevertheless, it remains possible to establish point-to-point correspondences between images and to reconstruct the surface along each line.

3.3 Shape from Contour

It is sometimes possible to extract a line drawing, such as the one shown in Figure 9, from scene textures. Parallel streets like those encountered in Figure 4 give rise to a virtual image consisting of parallel lines when the cross streets cannot be located; terraced hills also produce a virtual image of parallel lines. Correspondences between real and virtual image lines can be found by counting adjacent lines from an arbitrary starting point. This matches a virtual image line with each point in the real image. Point-to-line correspondences are not sufficient to enable the stereo computation of the appendix to be used for reconstruction of the surface. Knowledge of the relative orientation between the two images (equivalent to knowing the orientation of the camera that produced the real image relative to the parallel lines in the scene) provides an adequate constraint; the surface can then be reconstructed uniquely through back-projection. Without knowledge of the relative orientation of the virtual image, heuristics must be employed that relate points on adjacent contours so that a regular grid can be used as

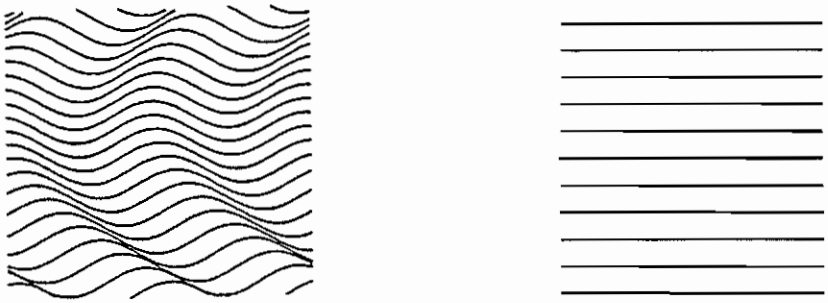

Figure 9: (a) An image of contours (b) Its virtual image

the virtual image. The human visual system is normally able to interpret images like Figure 9 unambiguously although just what assumptions are being made remains unclear. Further study of this phenomenon may make it possible to extract models that are especially suited to the employment of one-eyed stereo on this type of image without requiring prior knowledge of the virtual orientation.

3.4 Distorted Textures and Unfriendly Shading

We have already noted that image shading can be viewed as a limiting (and, for our purposes, a degenerate) result of closely spaced texture elements. To recover depth from shading, we must use integration instead of the process of counting the texture elements that define the locations of the "grid lines" of our virtual image. The integration process depends on the existence of a "friendly" reflectance function and an imaging geometry that allows us to convert distance along a line in the actual image to a corresponding distance along a line in the virtual image.

The recovery of lunar topography from a single shaded image [6], as discussed in Section 3.2, is one of the few instances in which "shape from shading" is known to be possible without a significant amount of additional knowledge about the scene. Nevertheless, even here we are required to know the actual reflectance function, the location of the [point] source of illumination, and the depths along a curve on the object surface, and be dealing with a portion of the surface that has constant albedo. Furthermore, the reflectance function has to have just the property we require to replace

direct counting, *i.e.*, the reflectance function has to compensate exactly for the "foreshortening" of distance due to viewing points on the object surface from any angle. Most of the commonly encountered reflectance functions, such as Lambertian reflectance, do not possess this friendly property, and it is not clear to what extent it is possible to recover depth from shading in such cases (*e.g.*, see Pentland [12] and Smith [15]). Additional assumptions will probably be necessary and the qualitative nature of the recovery will be more pronounced. Just as in the case in which a complex function can be evaluated by making a local linear approximation and iterating the resulting solution, so it may be possible to deal with unfriendly, or even unknown, reflectance functions by assuming that they are friendly in the vicinity of some point, solving directly for local shape by using the algorithm applicable to the friendly case, and then extending the solution to adjacent regions. We are currently investigating this approach.

The uniform rectangular grid and the polar grid that we used as virtual images to illustrate our approach to one-eyed stereo are effective in a large number of cases because there are processes operating in the real world that produce corresponding textures (*i.e.*, gridlike textures that appear to be orthographically projected onto the surfaces of the scene). However, there are also textures that produce similar-appearing images, but are due to different underlying processes. For example, a uniform gridlike texture might have been created on a flat piece of terrain that is subsequently subjected to geologic deformation—in this case the virtual image (or the recovery algorithm) needed to recover depth must be different from the projective case. We have already indicated the problem of choosing the appropriate model for the virtual image and, as noted above, image appearance alone is probably insufficient for making this determination—some semantic knowledge about the scene is undoubtedly essential. Figure 10 shows an example in which two completely different, yet equally believable, interpretations of scene structure result, depending on whether we use the rectangular grid model or the polar grid model.

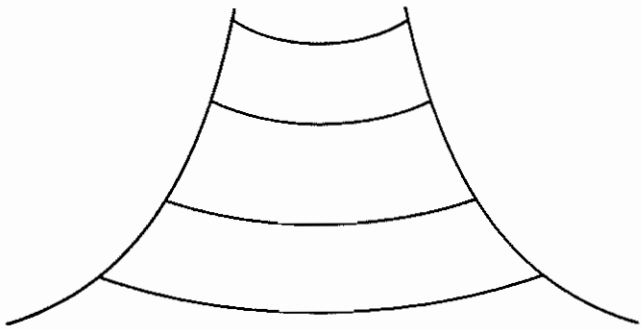

Figure 10: This simple drawing has two reasonable interpretations. It is seen as curved roller-coaster tracks if the lines are assumed to be the projection of a rectangular grid, or as a volcano when the lines are assumed to be the projection of a circular grid.

4 Experimental Results

The stereo reconstruction algorithm described in the appendix has been programmed and successfully tested on both real and synthetic imagery. Given a sparse set of image points and their correspondence in a virtual image, a qualitative description of the imaged surface can be obtained.

Synthetic images were created from surfaces painted with computer-generated graphic textures. Figure 11(a) shows a synthetic image constructed from a section of a digital terrain model (DTM). The intersections of every tenth grid line constitute the set of 36 image points made available to the one-eyed stereo algorithm. Their correspondences were established by selecting an arbitrary origin and counting grid lines to obtain virtual image coordinates. When these pairs are processed by the algorithm in the appendix, a set of 3-D coordinates is obtained in either the viewer-centered coordinate space, or the virtual image coordinate space (which, if correct, is aligned with the original DTM). Figure 11(b) was produced from the resulting 3-D coordinates expressed in the virtual image space by using Smith's surface interpolation algorithm [16] to fit a surface to these points. This yields a dense set of 3-D coordinates that can then be displayed from any viewpoint. The viewpoint that was computed by one-eyed stereo was used to render the surface as shown in Figure 11(b). Its similarity to the

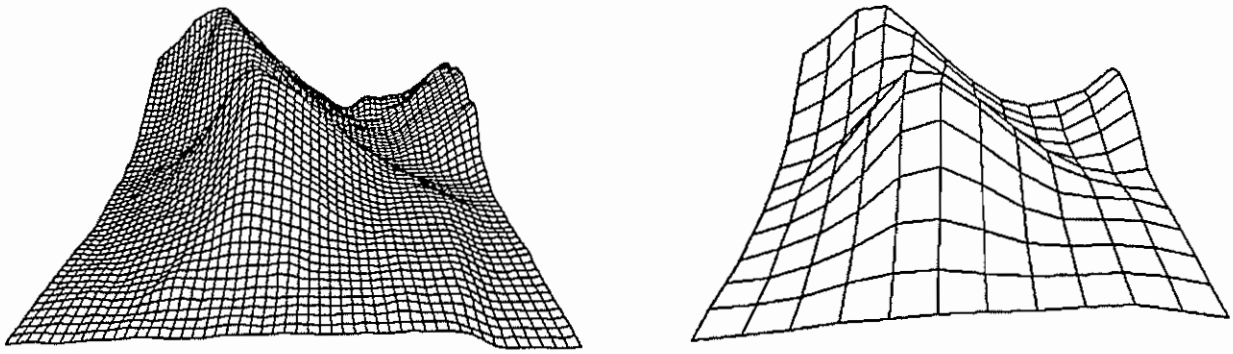

Figure 11: (a) View of part of a DTM (b) View of surface reconstructed from (a)

original rendering (Fig. 11(a)) confirms the successful reconstruction of the scene.

The same procedure was followed when we worked with real photographs. The intersections of 31 street intersections were extracted manually from the photograph of San Francisco shown in Figure 4. Those that were occluded or indistinct were disregarded. Virtual image coordinates were obtained by counting city blocks from the lower-left intersection. The one-eyed stereo algorithm was then used to acquire 3-D coordinates of the corresponding image points in both viewer-centered and grid-centered coordinate systems. A continuous surface was fitted to both representations of these points. The location and orientation of the camera relative to the grid were also computed. Figure 12(a) shows the reconstructed surface as an orthographic view from the direction computed to be true vertical. The numbers superimposed are the computed locations of the original 31 points. Figure 12(b) shows the surface from the derived location of the viewpoint of the original photo. While several of the original points were badly mislocated, the general shape of the landform is apparent.

There are several reasons the algorithm can provide only a qualitative shape description. First, the problem itself can be somewhat sensitive to slight perturbations in the estimates of the piercing point or focal length. This appears to be inherent to the problem of recovering shape from a single image. How humans can perceive shape monocularly without apparent

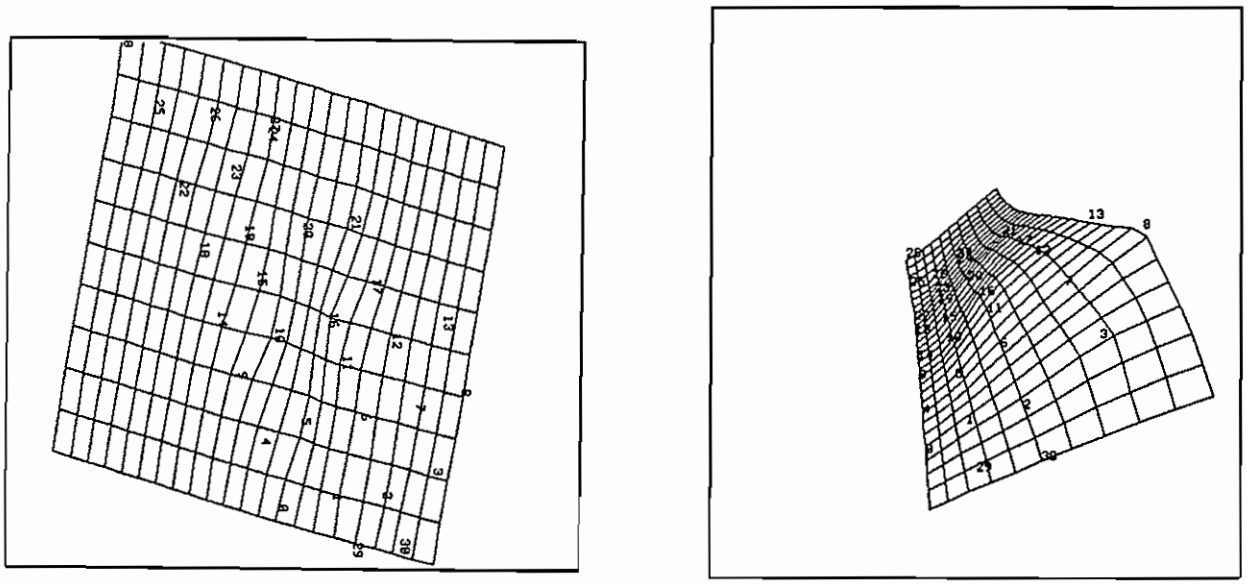

Figure 12: (a) Orthographic view of surface reconstructed from Figure 4 (b) Perspective view of same surface (from derived camera location)

knowledge of the piercing point or semantic content of the scene remains unresolved. The second factor precluding precise, quantitative description of shape is the practical difficulty of acquiring large numbers of corresponding points. While the algorithm can proceed with as few as eight points, the location of the object will be identified at those eight points only. If a more complete model is sought, additional points will be required to constrain the subsequent surface interpolation.

The task remains to evaluate the effectiveness of the iterative technique, described in Section 3.4, for recovering (1) shape from shading in the case of scenes possessing "unfriendly" reflectance functions, and (2) shape from nonprojective and distorted textures. Our experience with the process indicates that the key to surmounting these problems lies in the ability to establish valid correspondences with the virtual image. With these in hand, reconstruction of the surface can proceed as outlined in the foregoing discussion.

5 Conclusion

In this paper we have shown that, in principle, it is possible to employ the stereo paradigm in place of various approaches proposed for modeling 3-D scene geometry—including the case in which only one image is available. We have further shown that, for the case of a single image, the approach could be implemented by:

1. Setting up correspondences between portions of the image and some variants of a uniform grid, and;

2. Treating each image region and its grid counterpart as a stereo pair, and employing a stereo technique to recover depth. (We present a new algorithm that makes it possible to accomplish this step.)

3. Devising automatic procedures to partition the image, select the appropriate form of the virtual image, and establish the correspondences are all difficult tasks that were not addressed in this paper. Nevertheless, we have unified a number of apparently distinct approaches, that individually, also have to contend with these same pervasive problems (*i.e.,* partitioning, model selection, and matching).

References

[1] Barnard, S. T., and Fischler, M. A., "Computational Stereo," Computing Surveys, Vol. 14, No. 4, December 1982.

[2] Brady. M., ed., *Artificial Intelligence* (Special Volume on Computer Vision), Volume 17, Nos. 1–3, August 1981.

[3] Cameron, R., *Above San Francisco*, Cameron and Company, San Francisco, 1976.

[4] Ganapathy, S., "Decomposition of Transformation Matrices for Robot Vision," International Conference On Robotics, (IEEE Computer Society), Atlanta, Georgia, March 13-15, 1984, pp. 130–139.

[5] Gennery, D. B. "Stereo Camera Calibration," Proceedings of the IU Workshop, November 1979, pp. 101–107.

[6] Horn, B. K. P., "Image Intensity Understanding," MIT Artificial Intelligence Memo 335, August 1975.

[7] Kender, J. R., "Shape from Texture," Ph.D. thesis, Carnegie–Mellon University, CMU-CS-81-102, November 1980.

[8] Lawton, D. T., "Constraint-Based Inference from Image Motion," Proc. AAAI-80, pp. 31-34.

[9] Longuet-Higgins, H. C., "A Computer Algorithm for Reconstructing a Scene from Two Projections," Nature, Vol. 293, September 1981, pp. 133–135.

[10] Nagel, H., and Neumann, B., "On 3-D Reconstruction from Two Perspective Views," Proc. IEEE 1981.

[11] Nitzan, D., Bolles, R.C., it et. al., "Machine Intelligence Research Applied to Industrial Automation," 12th Report SRI Project 2996, January 1983.

[12] Pentland, A. P., "Shading into Texture" Proceedings AAAI-84, August 1984, pp. 269–273.

[13] Prazdny, K., "Motion and Structure from Optical Flow," Proc. IJCAI-79, pp. 704-704.

[14] Roach, J. W., and Aggarwal, J. K., "Determining the Movement of Objects from a Sequence of Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 554–562.

[15] Smith, G. B., "The Relationship between Image Irradiance and Surface Orientation," Proc. IEEE CVPR-83.

[16] Smith, G. B., "A Fast Surface Interpolation Technique," Proceedings: DARPA Image Understanding Workshop, October 1984, pp. 211–215.

[17] Stevens, K. A., "The Line of Curvature Constraint and the Interpretation of 3-D Shape from Parallel Surface Contours," AAAI-83, pp. 1057–1061.

[18] Stevens, K. A., "The Visual Interpretation of Surface Contours," Artificial Intelligence Journal Vol. 17, No. 1, August 1981, pp. 47–73.

[19] Strat, T. M., "Recovering the Camera Parameters from a Transformation Matrix," Proceedings: DARPA Image Understanding Workshop, October 1984, pp. 264–271.

[20] Tsai, R.Y. and Huang, T.S., "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-6, No. 1, Jan 1984, pp. 13-27.

[21] Ullman, S., *The Interpretation of Visual Motion*, The MIT Press, Cambridge, Mass., 1979.

[22] Witkin, A. P., "Recovering Surface Shape and Orientation from Texture," Artificial Intelligence Journal Vol. 17, No. 1, August 1981, pp. 17–45.

[23] Witkin, A., and Kass, M., "Analyzing Oriented Patterns," Proceedings IJCAI-85.

6 Appendix

The main body of this paper was devoted to showing how the problem of interpreting certain varieties of textured and shaded images can be transformed into equivalent problems in binocular stereo. Beginning with a perspective image, a second (virtual) image is hypothesized according to some presumed model of the original image. The model also specifies how to establish the correspondence between points in the two images. To compute the shape of the surfaces in the original scene, we need only compute the 3-D coordinates from the information in the two images, where the actual scene is a perspective projection and the virtual image has been constructed as an orthographic projection. This appendix shows how three-dimensional coordinates can be computed from point correspondences between a perspective and an orthographic projection when the relation between the imaging geometries is unknown.

We will use lowercase letters to denote image coordinates and uppercase letters for 3-D object coordinates. Unprimed coordinates will refer to the geometry of the perspective image, and primed coordinates to the orthographic image. Let x_1 and x_2 be the image coordinates of a point in the perspective image relative to an arbitrarily selected origin. Let $-d_1$ and $-d_2$ be the [unknown] image coordinates of the principal point and let f [> 0] be the focal length. The object coordinates associated with an image point are (X_1, X_2, X_3), where the origin coincides with the center of projection and the X_3 axis is perpendicular to the image plane. The X_3 coordinates of any object point will necessarily be positive.

The imaging geometry is as depicted in Figure 13 and yields the following standard perspective equations:

$$x_1 + d_1 = f\frac{X_1}{X3}; \qquad x_2 + d_2 = f\frac{X_2}{X_3} \tag{3}$$

For the orthographic image, x'_1 and x'_2 are the image coordinates (relative to an arbitrary origin) and (X'_1, X'_2, X'_3) is the world coordinate system defined such that

$$x'_1 = X'_1; \qquad x'_2 = X'_2 \tag{4}$$

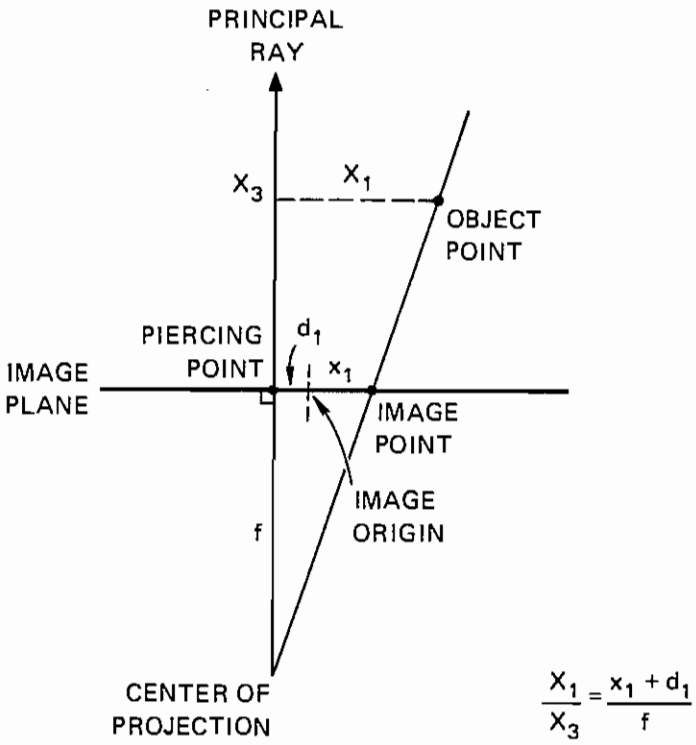

Figure 13: Definition of coordinate system

We use the unknown scale factor between orthographic image coordinates and the scene as our unit of measurement.

The two world coordinate systems can be related as follows:

$$X' = R(X - T) \quad , \tag{5}$$

where X is the column vector $[X_1, \ X_2, \ X_3]^T$,
X' is the column vector $[X'_1, \ X'_2, \ X'_3]^T$,
R is a 3x3 rotation matrix, and
T is a translation vector from the center of perspective projection to the origin of the world coordinate system associated with the orthographic projection. For either component (i=1 or 2), we can write

$$X'_i = R_i \cdot (X - T) \tag{6}$$

where R_i is the i-th row of R. By substituting Equations 3 and 4 into the above, we obtain

$$X_3 = \frac{f(x'_1 + R_1 \cdot T)}{R_1 \cdot [(x_1 + d_1), \ (x_2 + d_2), \ f]} \tag{7}$$

Eliminating X_3 from the two equations in Equation 7 yields

$$\begin{aligned} 0 = \ & x'_1 x_1 R_{21} + x'_1 x_2 R_{22} + x'_1 R_2 \cdot D - x'_2 x_1 R_{11} - x'_2 x_2 R_{12} - x'_2 R_1 \cdot D \\ & + x_1 (R_{21} R_1 \cdot T - R_{11} R_2 \cdot T) + x_2 (R_{22} R_1 \cdot T - R_{12} R_2 \cdot T) \\ & + R_1 \cdot T R_2 \cdot D - R_2 \cdot T R_1 \cdot D \end{aligned} \tag{8}$$

where D is the vector $[d_1, \ d_2, \ f]$.

The above equation relates image coordinates for corresponding points in both images. The following unknowns can be found by using eight corresponding pairs and solving the system of eight linear equations:

$$
\begin{aligned}
B_1 &= \tfrac{R_{21}}{R_{11}} \\
B_2 &= \tfrac{R_{22}}{R_{11}} \\
B_3 &= \tfrac{R_2 \cdot D}{R_{11}} \\
B_4 &= \tfrac{R_{12}}{R_{11}} \\
B_5 &= \tfrac{R_1 \cdot D}{R_{11}} \\
B_6 &= \tfrac{R_{21}}{R_{11}} R_1 \cdot T - R_2 \cdot T \\
B_7 &= \tfrac{R_{22}}{R_{11}} R_1 \cdot T - \tfrac{R_{12}}{R_{11}} R_2 \cdot T \\
B_8 &= \tfrac{1}{R_{11}} (R_1 \cdot T)(R_2 \cdot D) - \tfrac{1}{R_{11}} (R_2 \cdot T)(R_1 \cdot D)
\end{aligned}
\tag{9}
$$

$$
\begin{bmatrix}
x_1'x_1 & x_1'x_2 & x_1' & -x_2'x_2 & -x_2' & x_1 & x_2 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{bmatrix}
\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \end{bmatrix}
=
\begin{bmatrix} x_2'x_1 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}
\tag{10}
$$

When more than eight points are available, a least-squares method can be employed to solve the system of equations. Once we have the B_i's in hand, we can solve for the components of the rotation matrix R. First, R_{11} can be determined by making use of the fact that the rows of a rotation matrix are orthogonal. Thus, from $R_1 \cdot R_2 = 0$ and the expressions for B_1, B_2 and B_4 in Equation 9, we get

$$
R_{11}^4 (B_4^2 B_1^2 + B_2^2 - 2B_1 B_2 B_4) - R_{11}^2 (1 + B_1^2 + B_2^2 + B_4^2) + 1 = 0 \tag{11}
$$

This yields two real values for R_{11}; fortunately we'll be able to identify the incorrect one later. For now, let us simply choose one at random and return to this point if it turns out to be wrong.

The rest of R can be derived from the B_i's in a similar fashion. R_{12}, R_{21} and R_{22} can be established immediately from R_{11} and Equation 9. R_{13} is determined from the fact that $\| R_1 \| = 1$. $R_1 \cdot R_2 = 0$ gives an expression for R_{23}. Finally, R_3 is computed from the fact that $R_1 \times R_2 = R_3$ for all

rotation matrices. As a result, we have completely derived two alternative R matrices, depending on the choice of R_{11}. One of these matrices is correct, while the other can be eliminated later.

Now to solve for the translation vector T. First let us note that T cannot be found uniquely, because the origin of the primed world coordinate system has not been completely specified. The X'_1 and X'_2 coordinates of the origin were fixed by the choice of origin for the orthographic image coordinates, but the position of the origin along the X'_3 axis is still unconstrained. Since we are free to choose any origin for X', we will choose the one for which $T_3 = 0$.

Using the expression for B_6 in Equation 9, we find

$$B_6 = \frac{R_{21}}{R_{11}}(R_{11}T_1 + R_{12}T_2 + R_{13}T_3) - (R_{21}T_1 + R_{22}T_2 + R_{23}T_3) \tag{12}$$

Making use of the fact that $R_{33} = R_{11}R_{22} - R_{12}R_{21}$ and $T_3 = 0$, we get

$$T_2 = -B_6\frac{R_{11}}{R_{33}} \tag{13}$$

Similarly,

$$T_1 = B_7\frac{R_{11}}{R_{33}} \tag{14}$$

The origin of the primed coordinate system in unprimed coordinates is given by

$$T = [B_7\frac{R_{11}}{R_{33}},\ -B_6\frac{R_{11}}{R_{33}},\ 0]. \tag{15}$$

If the location of the principal point is known but the focal length (the scale factor of the perspective image) is not, f can easily be computed from Equation 9:

$$f = \frac{B_5R_{11} - R_{11}d_1 - R_{12}d_2}{R_{13}} \tag{16}$$

If the focal length is known, the principal point of the perspective image is found as follows. Use the third and fifth expressions of Equation 9 to write two equations in the two unknowns, d_1 and d_2. Their solution yields

$$\begin{aligned} d_1 &= f\frac{R_{31}}{R_{33}} + \frac{B_6R_{11}R_{22} - B_7R_{11}R_{12}}{R_{33}} \\ d_2 &= f\frac{R_{32}}{R_{33}} + \frac{B_7R_{11}^2 - B_6R_{11}R_{21}}{R_{33}} \end{aligned} \tag{17}$$

The perspective image coordinates of the principal point are $[-d_1, \ -d_2]$.

If neither the focal length nor principal point is known beforehand, then the problem we have proposed does not have a unique solution. Equation 17 specifies the constraints between focal length and piercing point. For any choice of focal length, there exists a unique principal point. The center of perspective projection is constrained to lie on a line parallel to the lines of sight of the orthographic projection. The reconstructed surface will be distorted as one varies the center of projection along this line. It is worth noting, however, that our computations of the rotation matrix R and the translation vector T did not require knowledge of either the focal length or the principal point.

We are now in a position to compute the world coordinates of all points for which we have correspondences. There may, of course, be many more than the minimum of eight points used so far. Equation 6 gives

$$x'_1 = R_1 \cdot [\frac{X_3}{f}(x_1 + d_1), \ \frac{X_3}{f}(x_2 + d_2), \ X_3] - R_1 \cdot T \tag{18}$$

which can be solved for

$$X_3 = \frac{f(x'_1 + R_1 \cdot T)}{R_{11}x_1 + R_{12}x_2 + R_1 \cdot D} \tag{19}$$

Now we must check the signs of the X_3's. If they are negative, the world points are located behind the center of projection. The correct solution, corresponding to all positive values of X_3, can be found by choosing the alternative value of R_{11} derived earlier and repeating the computations from that point. After obtaining the set of positive X_3's, we can continue.

Equation 3 gives the other unprimed world coordinates:

$$X_1 = \frac{X_3}{f}(x_1 + d_1); \qquad X_2 = \frac{X_3}{f}(x_2 + d_2) \tag{20}$$

If desired, the primed coordinates can be found by applying Equation 5.

The above derivation makes the implicit assumption that the X'_1 and X'_2 axes are scaled equally. It is conceivable that the virtual image coordinates could be unequally scaled, as is the case when they are derived from a rectangular grid (*e.g.,* Figure 4). If we have prior knowledge of the

ratio of the sides of each rectangular grid element, then the virtual image coordinates should be normalized before applying this algorithm (*i. e.*, by dividing X'_2 by this ratio). Without knowledge of the ratio, the problem is underspecified and a unidimensional class of solutions exists. Knowledge of the piercing point, if available, can be used to constrain the problem further and to solve for the unique solution. To do this, we define the following virtual coordinate systems in place of Equation (4):

$$x'_1 = X'_1; \qquad x'_2 = \frac{1}{k} X'_2 \tag{21}$$

where k is the ratio of the sides of the rectangular grid elements.

The solution proceeds as before, yielding

$$\begin{aligned} 0 = \; & x'_1 x_1 R_{21} + x'_1 x_2 R_{22} + x'_1 R_2 \cdot D - k x'_2 x_1 R_{11} - k x'_2 x_2 R_{12} - k x'_2 R_1 \cdot D \\ & + x_1 (R_{21} R_1 \cdot T - R_{11} R_2 \cdot T) + x_2 (R_{22} R_1 \cdot T - R_{12} R_2 \cdot T) \\ & + R_1 \cdot T R_2 \cdot D - R_2 \cdot T R_1 \cdot D \end{aligned} \tag{22}$$

The above equation is recast as the eight linear equations:

$$\begin{bmatrix} -x'_1 x_2 & -x'_1 & x'_2 x_1 & x'_2 x_2 & x'_2 & x_1 & x_2 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \end{bmatrix} = \begin{bmatrix} x'_1 x_1 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \tag{23}$$

where

$$\begin{aligned} C_1 &= \tfrac{R_{22}}{R_{21}} \\ C_2 &= \tfrac{R_2 \cdot D}{R_{21}} \\ C_3 &= \tfrac{k R_{11}}{R_{21}} \\ C_4 &= \tfrac{k R_{12}}{R_{21}} \\ C_5 &= \tfrac{k R_1 \cdot D}{R_{21}} \\ C_6 &= \tfrac{R_{11}}{R_{21}} R_2 \cdot T - R_1 \cdot T \\ C_7 &= \tfrac{R_{12}}{R_{21}} R_2 \cdot T - \tfrac{R_{22}}{R_{21}} R_1 \cdot T \\ C_8 &= \tfrac{1}{R_{21}} (R_2 \cdot T)(R_1 \cdot D) - \tfrac{1}{R_{21}} (R_1 \cdot T)(R_2 \cdot D) \end{aligned} \tag{24}$$

The following equalities can then be derived from Equation (24):

$$\begin{aligned} R_{13} &= \tfrac{R_{21}}{fk}(C_5 - C_3 d_1 - C_4 d_2) \\ R_{23} &= \tfrac{R_{21}}{f}(C_2 - d_1 - C_1 d_2) \end{aligned} \tag{25}$$

$$f = \sqrt{\frac{-(C_5 - C_3 d_1 - C_4 d_2)(C_2 - d_1 - C_1 d_2)}{C_3 + C_1 C_4}} \tag{26}$$

$$R_{21} = \pm \frac{f}{\sqrt{f^2 + C_1^2 f^2 + (C_2 - d_1 - C_1 d_2)^2}} \tag{27}$$

$$k = \frac{R_{21}}{f}\sqrt{f^2 C_3^2 + f^2 C_4^2 + (C_5 - C_3 d_1 - C_4 d_2)^2} \tag{28}$$

The rest of R can now be computed easily from Equation (24) and $R_1 \times R_2 = R_3$. The translation vector T is given by Equation (15) because $C_6 = B_6$ and $C_7 = B_7$. With R and T now fully recovered, it is a simple matter to derive the object coordinates from Eqs. (3), (21), and (5). Let us recall that we have two candidate matrices R hinging on the choice for R_{21}; as before, the correct one must be selected by examining the signs of the X_3 coordinates.

To summarize, we have described an algorithm to compute the relative orientation and position between two imaging systems—perspective and orthographic—from the locations of eight (or more) corresponding image points. Either the principal point or the focal length and rectangular aspect ratio are computed along the way. With this information in hand, the world coordinates of all points in the imaged scene can be computed.