line. 3) Images more complicated then a single line and background noise can be handled by similar methods. 4) Since the problem is related to a mixture of two Gaussians, an analytical solution may be possible. One can also associate directions with the points, as in [1].

ACKNOWLEDGMENT

The author thanks Prof. A. Rosenfeld for very helpful discussions.

REFERENCES

[1] A. Rosenfeld and A. Sher, "Direction-weighted line fitting to edge data," Comput. Vision Lab., Univ. Maryland, College Park, Tech. Rep. CAR-TR-189, 1986.

3D Euclidean Versus 2D Non-Euclidean: Two Approaches to 3D Recovery from Images

KEN-ICHI KANATANI

***Abstract*—Methods of 3D recovery in computer vision for computing the shape and motion of an object from projected images when an object model is available are classified into two types: the "3D Euclidean approach" based on geometrical constraints in 3D Euclidean space and the "2D non-Euclidean approach" based on analysis on the image plane viewed as a 2D non-Euclidean space. Implications of these two approaches are discussed, and some illustrating examples are presented.**

***Index Terms*—computer vision, image analysis, image understanding, 3D reconstruction, 3D recovery, shape from · · ·**

I. IMAGE UNDERSTANDING AND 3D RECOVERY

When we talk about "image understanding," the term "understanding" usually implies *modeling* of the objects because reconstructions of 3D raw data—called *3D images*—by such direct measurements as laser or ultrasonic ranging, stereo, and computer tomography (CT)—cannot be called "understanding"; a 3D image is only a collection of data in a three-dimensional array, just as a 2D image is a collection of data in a two-dimensional array. It is not until a model is fitted that we can say something about the object.

A model is specified by a small number of parameters. For example, an object can be modeled by a line, a plane, a quadric surface, a sphere, a cylinder, a cone, or a combination of these (e.g., a polyhedron). A line is specified by a point and the unit vector along it, a plane by a point and the unit normal, a quadric surface by the coefficients of the defining equation, a sphere by its center and radius. The 3D position of an object is specified by the position vector. The 3D orientation of an object is specified by three mutually orthogonal vectors starting from a fixed point on the object (or rotation matrix, Euler angles, quaternion, etc.). If the object is in motion, its 3D motion is specified by a translation velocity at a reference point and a rotation velocity (given by, say, an axis and an angular velocity) around it. Let us call these parameters which specifying the object shape, position, orientation, and motion *object parameters.*[1]

On the other hand, let us call the *numerical data* characterizing the observed image *image characteristics.*[2] They may reflect the gray-levels of the image, the texture of the object surface image, the object contour, the intensity of light reflection, or the optical flow if the object is in motion. Stated in this way, the 3D recovery problem is viewed as *estimation of the object parameters from the image characteristics.* This problem is often referred to as *shape from* · · · according to the source of the image characteristics—*shape from texture*, *shape from shading*, and *shape from motion*, for example. Therefore, the 3D recovery problem can be solved if equations relating the object parameters and the image characteristics are obtained. Let us call such equations *3D recovery equations.* It is desirable that the 3D recovery equations have simple forms, hopefully yielding analytical solutions. Otherwise, solutions must be computed numerically, which often results in difficulties—multiple solutions may exist, iterations may not converge, or computation time may be too long. Therefore, it is very important to obtain *good* 3D recovery equations. All subsequent processes are affected by the choice of the 3D recovery equations.

II. THE 3D EUCLIDEAN APPROACH

One approach to obtain 3D recovery equations is to treat all quantities *in 3D space.* First, an image itself is regarded as a 3D object by setting the image plane in the scene according to the camera model. Next, the image is *backprojected* into the scene by introducing unknown parameters (Fig. 1). For example, a point P on the image plane is backprojected into the point whose position vector is $r\overrightarrow{OP}$ where O is the viewpoint and r is an unknown depth parameter. Similarly, a line l on the image plane is backprojected onto a line lying somewhere on the plane defined by the viewpoint O and the line l.[3] In this way, starting from observed data, we can construct a *family of infinitely many candidates* of the object shape in such a way that all of them yield the same image data after perspective projection. Then, one solution is selected. The selection is done according to the *constraints* on the object. We select, from among the infinitely many backprojected candidates, the one which possesses properties that the object model is required to satisfy.

Object constraints are usually expressed in terms of the 3D Euclidean metric—a particular line segment of object must have a fixed length, two line segments must make a fixed angle or intersect perpendicularly, some line segments must be coplanar, or all lengths and angles are preserved during a rigid motion, for example. This means that the object constraints themselves play the role of 3D recovery equations. We call this approach the *3D Euclidean approach* because 3D recovery equations have geometrical meanings in *3D Euclidean space*; they may specify conditions concerning lengths, angles, parallelism, orthogonality, planarity, rigidity, etc. They are usually expressed as 3D vector and matrix equations, and subsequent analyses are done in terms of *vector calculus* and *matrix algebra.*

A great majority of past research on 3D recovery has been done with this approach, since it is very natural to ask what family of objects can be projected onto the observed image, and then choose the one that best fits our prior knowledge about the object. In fact, this viewpoint has long been adopted by psychologists in the study of human visual perception (e.g., Gibson [3]), and although psy-

Manuscript received February 19, 1987; revised November 25, 1988. Recommended for acceptance by W. B. Thompson. This work was supported in part by the Casio Science Promotion Foundation, Yazaki Memorial Foundation for Science and Technology, Inamori Foundation, and the Japanese Ministry of Education, Science, and Culture under Grant in Aid for Scientific Research (63550268).

The author is with the Department of Computer Science, Gunma University, Kiryu, Gunma 376, Japan.

IEEE Log Number 8825721.

[1]In the literature, they are called by many other names—*structure and motion parameters*, for example.

[2]Some authors call such data *image features*, *image properties*, or *observables.*

[3]However, such backprojection is possible only if the point P or the line l corresponds to a point or a line in the scene. This is one of the difficulties of this approach, as we will discuss later.

0162-8828/89/0300-0329$01.00 © 1989 IEEE

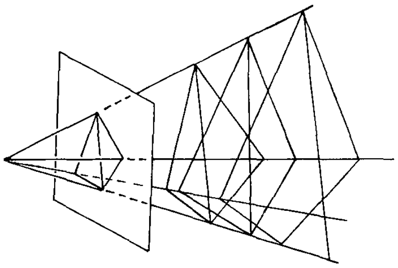

Fig. 1. *Backprojection.* A family of infinitely many object shapes are constructed such that they all yield the same image after projection. Then, the one which satisfies the required *constraints* is selected.

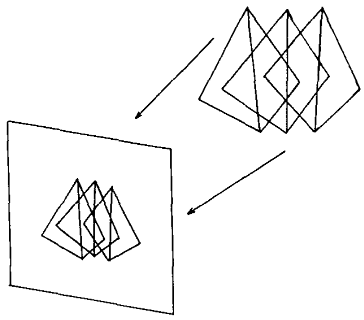

Fig. 2. A parameterized 3D object model is projected onto the image plane, resulting in a parameterized family of infinitely many images. Then, the one which best matches the observed image is selected. The matching is done by comparing a small number of *image characteristics.*

chologists are not so very interested in the computational aspects, their approaches have exerted a great influence over studies of computer vision (e.g., Ullman [22], Marr [13]).

A major disadvantage of this approach is that the resulting 3D recovery equations become very complicated. Often, analytical solutions are difficult to obtain. One reason is that image characteristics and object parameters are often chosen so as to make the backprojection easy. For example, backprojection is easy if the image data have straightforward meanings like positions, orientations, lengths, and angles, but it is difficult to backproject global quantities such as sums and averages. The object parameters, on the other hand, also tend to be quantities describing the object *relative to the image plane*, such as its distance from the image plane. Hence, they may not necessarily indicate properties inherent to the object such as the surface gradient and surface curvature.

Another point to note is that this approach results in 3D vector and matrix equations, which are easy to analyze *if the X-, Y-, and Z-axes play interchangeable roles.*[4] In the 3D recovery problem, however, the Z-axis, taken to coincide with the camera optical axis, plays a special role. This means that the three components do not have the same geometrical meaning; they only have a symmetry with respect to the X- and Y-components. Is there any way to exploit this fact explicitly?

III. The 2D Non-Euclidean Approach

The 3D Euclidean approach starts with an observed 2D image and ends with constraints on the 3D object model. An alternative approach is to start with a *parameterized 3D object model.* Since the imaging geometry is simple, it is easy to compute the expected projection image. This process, called *projection* as opposed to *backprojection*, defines a *parameterized family of infinitely many images* (Fig. 2). These images are regarded as *2D quantities*; their 3D origins can be ignored.

Once a family of 2D images is defined, the next step is to *define* image characteristics. Since the images are parameterized, the image characteristics thus defined are functions of object parameters. Then, we turn to the observed image. The 3D recovery equations are obtained by measuring the corresponding image characteristics on the observed image and equating their values with their theoretically predicted expressions.

Since we start with an object model, we can choose object parameters in such a way that they have desirable properties. It is desirable that the parameters have geometrical meanings inherent to the object itself and independent of the choice of the coordinate system.[5] The ease of subsequent analysis is affected by the choice of these parameters.[6] Thus, this approach allows more freedom and flexibility than the 3D Euclidean approach where the choice of object parameters is virtually dictated by the convenience of backprojection.

After the object parameters are chosen, there is still another choice to make. We must define *good* image characteristics which are capable of distinguishing images with different object parameter values. Since no coordinate system is inherent to the images, the image x- and y-axes must play interchangeable roles. Furthermore, if another $x'y'$-coordinate system is taken by rotating the original xy-coordinate system around the image origin, these two must play equivalent roles. Hence, the image characteristics should be symmetric with respect to the x- and y-coordinates and have *the same geometrical meaning* if the coordinate system is rotated. In this sense, *3D recovery is possible if and only if the image characteristics have coordinate rotation invariance* [6]. The fact that the image characteristics are *defined*, as opposed to *given* as for the 3D Euclidean approach, provides great freedom and flexibility in analyzing 3D recovery problems. In particular, image characteristics defined as *linear functionals* (e.g., weighted sums or averages of image data) play important roles in practical applications.

A line in the scene is projected onto a line on the image plane, but projections of parallel or orthogonal lines are generally no longer parallel or orthogonal on the image plane *if measured in the 2D Euclidean metric of the image plane.* Also, the length of a line segment in the scene is not preserved by projection *if measured in the 2D Euclidean metric of the image plane.* The distortion due to projection differs from position to position *if measured in the 2D Euclidean metric of the image plane.* However, *if we assume an object model*, we can define parallelism and orthogonality on the image plane which reflect parallelism or orthogonality in the scene by introducing an appropriate *2D non-Euclidean metric.* It is possible to define a 2D non-Euclidean metric on the image plane in such a way that the lengths and angles measured in that metric have a geometrical meaning in the original 3D space. Thus, the image plane can be regarded as a *2D non-Euclidean space.* For example, if the object is modeled as a plane, the Euclidean geometry on it is converted into *2D projective geometry* on the image plane. From these observations, let us call the approach described in this section the *2D non-Euclidean approach.*

IV. 3D Euclidean Versus 2D Non-Euclidean

The 3D Euclidean approach begins with image characteristics on the image plane, *backprojects* them into the scene, and applies *object constraints* expressed in terms of 3D Euclidean geometry.

[4]If we define an *image sphere* surrounding the viewpoint and project the scene onto it, each point has the same role. However, an xy-image plane is the most convenient for many practical treatments. Thus, as long as we use an xy-image plane, we must take into account its special geometry.

[5]For example, a plane in the scene is specified either by the 3D coordinates of three points on it or the pair of its surface gradient and depth. Evidently, the latter choice catches the essential meaning of the plane.

[6]For example, the shape of a corner defined by three faces is specified by either the surface gradients of the three faces (the *gradient space representation*) or the 3D orientations of the three edges. The latter choice is more convenient for 3D recovery from angle clues of project images [8].

The mathematical tools are 3D vector calculus and matrix algebra. In contrast, the 2D non-Euclidean approach begins with *object modeling*, then *projects* the model onto the image plane, and *defines* image characteristics in terms of the 2D non-Euclidean geometry resulting from the assumed object model. As a result, various mathematical tools become available—*differential geometry* and *tensor calculus*, for example. Moreover, the 2D non-Euclidean approach can exploit various *invariance properties* over some *groups of transformations*. The geometrical interpretation of image characteristics should be invariant to coordinate rotations [6], and the information contained in an image is preserved by the camera rotation [7]-[9].

If the object satisfies some constraints (e.g., collinearity, coplanarity, parallelism, and orthogonality), they give rise to *consistency conditions* that the projected object image must satisfy. If a given image does not satisfy the consistency conditions, that image is a *false image*, i.e., it cannot be obtained by projecting a real object. Due to the existence of noise, however, real images often do not exactly satisfy the consistency conditions even if they are projections of real objects. As a result, if we try to backproject them, the candidates to be constructed in the 3D scene may be empty or may not contain the true solution. This inconsistency can be easily overcome by the 2D non-Euclidean approach because essentially what it does is *matching* on the image plane. Moreover, we need not seek an *exact match*; we can seek only for the *best match*.

The 2D non-Euclidean approach does not directly match images; *matching is done at the level of the image characteristics*, and other image properties are ignored. In other words, our choice of image characteristics *defines* the matching, and the 3D recovery equations are viewed as the *matching conditions*. Robustness to noise can be increased if we choose, as image characteristics, global quantities such as sums or averages of a large number of measured values. It is very difficult to backproject such global quantities meaningfully.

Another advantage of the 2D non-Euclidean approach is that we *need not* identify the three-dimensional meanings of the image characteristics; the image characteristics are defined as purely 2D properties of the image (e.g., the average intensity, the area inside the object contour, etc.). If the object is in motion, its 3D motion can be determined without detecting *point-to-point correspondences*, i.e., without requiring knowledge of which point corresponds to which between two image frames.

However, the distinction between the 3D Euclidean and the 2D non-Euclidean approaches is sometimes not so clear-cut.[7] There also exist hybrid approaches to mix the two approaches. Note that the distinction between the two approaches lies in the interpretation of whether the 3D recovery equations are regarded as *3D constraints* or *2D matching conditions*.

V. Examples and Concluding Remarks

A typical problem for which the distinction is very clear is the detection of 3D motion from image motion (*i.e., shape from motion*). Suppose we observe a motion of n points on the image plane. Let (x_i, y_i), $i = 1, \cdots, n$, be the positions of the n points on the image plane, and let $(\dot{x}_i, \dot{y}_i)$, $i = 1, \cdots, n$, be their image velocities. We want to know where these points are located in the scene and how they are moving three dimensionally. Let us assume, for simplicity, that the projection is orthographic.

First, let us try the 3D Euclidean approach. If we introduce, as unknowns, the depth z_i and the velocity component $\dot{z}_i$ along the z-axis for each point, we can describe the 3D motion of these points in the scene in terms of image characteristics $x_i, y_i, \dot{x}_i, \dot{y}_i$, $i = 1, \cdots, n$. Namely, we obtain a *backprojected 3D description* that a point (x_i, y_i, z_i) is moving with velocity $(\dot{x}_i, \dot{y}_i, \dot{z}_i)$, $i = 1, \cdots, n$. In order to determine the unknowns $z_i, \dot{z}_i$, $i = 1, \cdots, n$, we need constraints. Suppose the only knowledge we have is that they are moving *rigidly*. This constraint is written as $(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 = \text{const.}$ for $i \neq j$. Differentiating this, we have

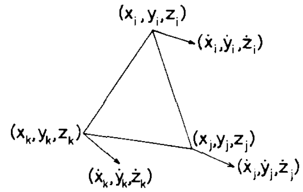

Fig. 3. The observed motion of points on the image plane is backprojected into the scene and regarded as a 3D motion if the depth z_i and the velocity $\dot{z}_i$ along the Z-axis are introduced for each point. These unknown parameters are determined by the constraint that the resulting 3D motion should be rigid.

$$(x_i - x_j)\dot{x}_i + (y_i - y_j)\dot{y}_i + (z_i - z_j)\dot{z}_i = 0 \qquad (1)$$

for $i \neq j$ (Fig. 3). These equations are the only equations constraining the unknowns, $z_i, \dot{z}_i$, $i = 1, \cdots, n$. It can be shown that numerical solutions are obtained if we introduce three indeterminate parameters. This is the approach taken by Sugihara and Sugie [18].

However, this approach produces only a *numerical algorithm* of computing $z_i, \dot{z}_i$, $i = 1, \cdots, n$, for given values of the indeterminate parameters; no closed-form analytical solution is available, and hence no theoretical properties and behaviors of the solution are obtained. Moreover, it turns out that the solutions computed by this approach contain a *false solution*. Namely, although (1) must hold for *any* rigid motion, it also holds for *some* nonrigid motions. As a result, the numerical solutions computed from (1) also contain a nonrigid motion solution.

Let us try a 2D non-Euclidean approach. In contrast to the previous approach, we start with the 3D description that n points are moving in the scene. Consider three points among them. A plane or *planar patch* is defined by the three points. A plane rigidly moving in the scene induces an *optical flow* on the image plane (Fig. 4). It is easy to prove that the flow is described by linear equations in the form

$$\dot{x} = a + Ax + By, \quad \dot{y} = b + Cx + Dy. \qquad (2)$$

The form of these equations is invariant to rotations of the image xy-coordinate system because, however the coordinate system is taken, we are still viewing a rigid motion of a plane. Hence, if the xy-coordinate system is rotated, (2) is again obtained with different values of the coefficients a, b, A, B, C, D. From this fact, we can construct *invariants* from these coefficients,[8] and obtain a closed-form solution in terms of them. Thus, we only need to measure the six coefficients (i.e., the *image characteristics*) a, b, A, B, C, D on the image plane.

At this very last stage, we consider how to estimate (or *match*) the six image characteristics from the given data (x_i, y_i), $(\dot{x}_i, \dot{y}_i)$, $i = 1, \cdots, n$. From (2), we see that if the velocity components $\dot{x}, \dot{y}$ are given at three points which define a planar patch, the six image characteristics a, b, A, B, C, D are determined simply by solving a set of six simultaneous linear equations. This process is applied to the rest of the planar patches by choosing appropriate three points repeatedly. This is the approach of Kanatani [4], and

[7] The distinction between the two approaches is less clear for orthographic projection than for perspective projection because many 2D quantities under orthographic projection are at the same time thought of as 3D quantities.

[8] An *invariant* does not necessarily have a constant value, i.e., it need not be an *absolute invariant*. It can change its value as long as the change depends on its own value but not on other quantities, i.e., it can be a *relative invariant* [6], [7], [9].

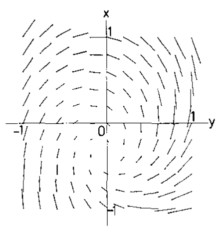

Fig. 4. The *optical flow* on the image plane resulting from a planar surface motion is analyzed, and the object motion is determined in such a way that the expected flow is compatible with the actually observed motion of points.

he deduced many interesting properties and behaviors of the solution.

The problem can be extended to perspective projection, and many researchers have obtained solutions for finite motions under various conditions. Most of them used the 3D Euclidean approach, describing and analyzing the problem in terms of 3D vector and matrix calculus in 3D Euclidean space [11], [14], [20]. For instantaneous motions, however, the 2D non-Euclidean approach has been used more effectively [5], [12], [19].

Motion detection is closely related to 3D recovery of object position and orientation. According to the 3D Euclidean approach, the position and orientation of an object whose 3D shape is known are computed from a projected image by first backprojecting the image edges into the scene, and then applying the constraint that the resulting 3D shape should coincide with a known shape [16]. According to the 2D non-Euclidean approach, on the other hand, an object model is first placed in the scene, and then its 2D image projected onto the image plane is analyzed. The object position and orientation are estimated by matching image characteristics (or *features*, *observables*, etc.) [1] [10]. Since these quantities are defined globally over the object image, no knowledge of the correspondence between the vertices and edges of the image and those of the object model is required.

Another problem for which the two approaches can be contrasted is the recovery of 3D road geometry from images for purposes of navigating *autonomous land vehicles* (ALV's). A typical 3D Euclidean approach was presented by DeMenthon [2], who backprojected the observed road boundary image into the scene and searched for the solution that satisfies the constraints the real roads are supposed to obey. In contrast, Thorpe *et al.* [19] first assumed a parameterized straight and horizontal road and then determined the parameters by matching the resulting projection image and the actually observed image. Sakurai *et al.* [15] used curved and nonhorizontal road models with a similar approach. Turk *et al.* [21] called the 3D Euclidean approach and the 2D non-Euclidean approach the *forward geometry* and the *backward geometry*, respectively.[9]

Other topics of 3D recovery where mathematics is involved include computation of angles in the scene from angles observed on the image plane, and computation of surface gradient from observed texture density (i.e., *shape from texture*). Although it is difficult to say which author used which approach, some authors clearly followed the reasoning based on the 3D Euclidean approach, while others seemed to be strongly influenced by the thinking of the 2D non-Euclidean approach. In this paper, the significance of the 2D non-Euclidean approach has been emphasized, but of course, the choice depends on the problem to be solved; the 3D Euclidean approach may be very useful for some problems, and a hybrid approach may be fruitful for other problems. In any case, we can understand various approaches better if we always keep in mind the contrast of the two typical approaches of 3D recovery.

[9]Other approaches include detection of vanishing points [23], which is difficult to classify into either of them.

References

[1] T.-C. Chou and K. Kanatani, "Recovering 3D rigid motions without correspondence," in *Proc. IEEE 1st Int. Conf. Comput. Vision*, London, England, June 1987, pp. 534–538.

[2] D. DeMenthon, "A zero-bank algorithm for inverse perspective of a road from a single image," in *Proc. IEEE Int. Conf. Robotics Automation*, Raleigh, NC, Apr. 1987, pp. 1444–1449.

[3] J. J. Gibson, *The Ecological Approach to Visual Perception.* Boston, MA: Houghton Mifflin, 1979.

[4] K. Kanatani, "Structure and motion from optical flow under orthographic projection," *Comput. Vision Graphics Image Process.*, vol. 35, pp. 181–199, 1986.

[5] —, "Structure and motion from optical flow under perspective projection," *Comput. Vision Graphics Image Process.*, vol. 38, pp. 122–146, 1987.

[6] —, "Coordinate rotation invariance of image characteristics for 3D shape and motion recovery," in *Proc. IEEE 1st Int. Conf. Comput. Vision*, London, England, June 1987, pp. 55–64.

[7] —, "Camera rotation invariance of image characteristics," *Comput. Vision Graphics Image Process.*, vol. 39, pp. 328–354, 1987.

[8] —, "Constraints on length and angle," *Comput. Vision Graphics Image Process.*, vol. 41, pp. 28–42, 1988.

[9] —, "Transformation of optical flow by camera rotation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 131–143, Feb. 1988.

[10] K. Kanatani and K. Yamada, "Model-based determination of object position and orientation without matching," *J. Inform. Process.*, vol. 12, no. 1, 1989.

[11] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 10, pp. 133–135, 1981.

[12] H. C. Longuet-Higgins, "The visual ambiguity of a moving plane," *Proc. Roy. Soc. Lond.*, vol. B-223, pp. 165–175, 1984.

[13] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* San Francisco, CA: Freeman, 1982.

[14] H. H. Nagel, "Representation of moving objects based on visual observations," *Computer*, vol. 14, no. 8, pp. 29–39, 1981.

[15] K. Sakurai, H. Zen, H. Ohta, Y. Ushioda, and S. Ozawa, "Analysis of a road image as seen from a vehicle," in *Proc. IEEE 1st Int. Conf. Comput. Vision*, London, England, June 1987, pp. 651–656.

[16] T. M. Silberberg, D. Harwood, and L. S. Davis, "Object recognition using oriented model points," *Comput. Vision Graphics Image Process.*, vol. 35, pp. 47–71, 1986.

[17] M. Subbarao and A. M. Waxman, "Closed form solutions to image flow equations for planar surface in motion," *Comput. Vision Graphics Image Process.*, vol. 36, pp. 208–228, 1986.

[18] K. Sugihara and N. Sugie, "Recovery of rigid structure from orthographically projected optical flow," *Comput. Vision Graphics Image Process.*, vol. 27, pp. 309–320, 1984.

[19] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, pp. 362–373, 1988.

[20] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 6, pp. 13–27, Jan. 1984.

[21] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "VITS—A vision system for autonomous land vehicle navigation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, pp. 342–361, Mar. 1988.

[22] S. Ullman, *The Interpretation of Visual Motion.* Cambridge, MA: M.I.T. Press, 1979.

[23] A. M. Waxman, J. J. LeMoigne, L. S. Davis, B. Srinivasan, T. R. Kushner, E. Liang, and T. Siddalingaiah, "A visual navigation system for autonomous land vehicles," *IEEE J. Robot. Automat.*, vol. 3, pp. 124–141, 1987.