A SIMPLE SENSOR TO GATHER THREE-DIMENSIONAL DATA

Technical Note 249

July 17, 1981

By: Robert C. Bolles, Senior Computer Scientist
Jan H. Kremers, Computer Scientist
Ronald A. Cain, Systems Programmer

Industrial Automation Department
Computer Science and Technology Division

SRI Project 1538

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 859-6200 • TWX: 910-373-2046 • Telex: 334 486

ABSTRACT

A simple triangulation-based range sensor and its calibration procedure are described. The sensor consists of a projector that produces a plane of light, a camera that observes the intersection of the plane of light and objects in the scene, and a computer that calculates the x-y-z positions of points along the intersection. Sensors of this type have been used in several research laboratories, but the mathematical characterization of these sensors has not been widely disseminated. In this paper straighforward procedures are presented for calibrating a camera, computing the equation of a plane, and combining a camera calibration and an equation of a light plane to form a "sensor" matrix. The sensor matrix can be used to compute the x-y-z positions for points along the intersection efficiently.

CONTENTS

APPENDICES

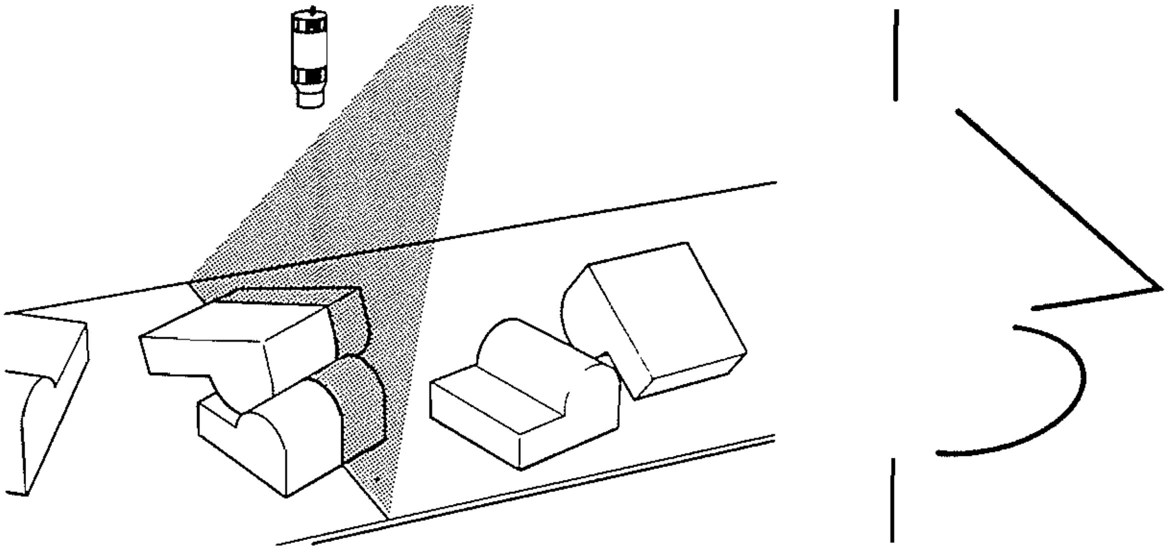
ILLUSTRATIONS

I INTRODUCTION

Interpreting a two-dimensional image of a three-dimensional scene is difficult for a computer for two reasons. First, the three-dimensional information is compressed into two dimensions; second, the intensities in the image are complex functions of several factors--such as the positions and orientations of the surfaces in the scene, the reflectances of the surfaces, and the positions and intensities of the light sources. People overcome these difficulties in complex, imperfectly understood ways--including stereo processing and a well-developed set of models that anticipate what exists in the world and how it should appear. Although progress has been made toward emulating this type of processing in a computer [1-6], practical computer vision systems for the foreseeable future will either avoid three-dimensional ambiguities by concentrating on two-dimensional tasks or will use special sensors to extract three-dimensional information directly.

In this paper we describe a sensor that uses a camera and a projected plane of light to gather three-dimensional information (see Figure 1). This type of sensor has been used in several laboratories [7-11], but the supporting mathematics is not widely known outside these institutions. We at SRI have used such a sensor to inspect three-dimensional parts [12], provide position and orientation feedback for a robot arm [10], and gather range descriptions of stacked industrial parts [13].

The sensor uses triangulation to compute x-y-z positions for points on the sensed objects. Triangulation is required because it is not possible to determine the x-y-z position of a point from just its image position. It is only possible to determine a ray in space on which the point must lie. This limitation is due to the fact that information is lost in the projection of a three-dimensional scene onto a two-



(a) DESIGN OF A RANGE SENSOR BASED ON A CAMERA AND A PLANE OF LIGHT

(b) IMAGE OF THE INTERSECTION OF THE LIGHT PLANE AND THE OBJECTS

FIGURE 1 A SIMPLE RANGE SENSOR

dimensional image. To overcome this limitation the sensor uses a plane of light to locate a unique point along each ray in space. Considered in another way, the reason the sensor can compute three-dimensional information is that it is possible to establish a one-to-one correspondence between points in the image and points in the light plane. Knowing the position of the light plane in some coordinate system, such as that of the world or of the camera, makes it possible to convert positions on the plane to x-y-z positions in that coordinate system.

In this paper we represent a camera's projective transformation by a 4 X 4 "camera" matrix that maps three-dimensional world coordinates into two-dimensional image coordinates (see Figure 2). Both sets of coordinates are represented in homogeneous coordinates (see Appendix A for an introduction to the latter). The mapping is as follows:

$$\begin{pmatrix} s*u \\ s*v \\ s*w \\ s \end{pmatrix} = \begin{pmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \\ a41 & a42 & a43 & a44 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} , \tag{1}$$

where (x,y,z) is a world point, the aij's are the components of the camera matrix, and (u,v) is the corresponding image point. This equation implies the following expressions for the coordinates of the image point:

$$u = \frac{a11*x + a12*y + a13*z + a14}{a41*x + a42*y + a43*z + a44} \tag{2}$$

$$v = \frac{a21*x + a22*y + a23*z + a24}{a41*x + a42*y + a43*z + a44} \quad . \tag{3}$$

The ray in space defined by an image point and the lens center can be represented as the intersection of two planes, the u and v planes in Figure 3. The equations of these can easily be derived from the expressions for u and v:

$$(a11-u*a41)*x + (a12-u*a42)*y + (a13-u*a43)*z + (a14-u*a44) = 0 \tag{4}$$

$$(a21-v*a41)*x + (a22-v*a42)*y + (a23-v*a43)*z + (a24-v*a44) = 0. \tag{5}$$

As mentioned above, it is not possible, without some additional information, to determine uniquely the three-dimensional position of a point in the world corresponding to a point in an image. One adequate piece of information is provided if the point in the world is known to lie in some plane in the world. Then the point's three-dimensional position is uniquely determined as the intersection of three planes: the two defined by its position in the image and the one known a priori (see Figure 3). In particular, if the equation of the a priori plane is

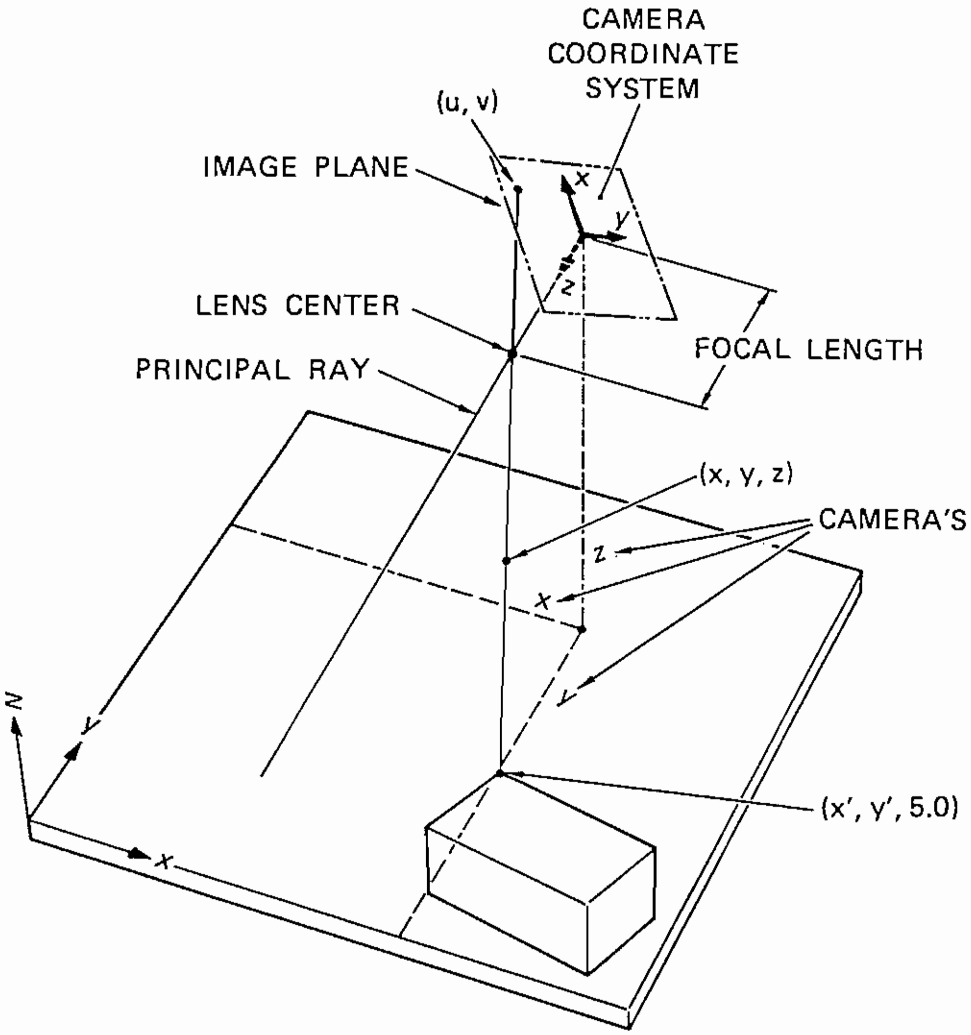


FIGURE 2 CAMERA GEOMETRY

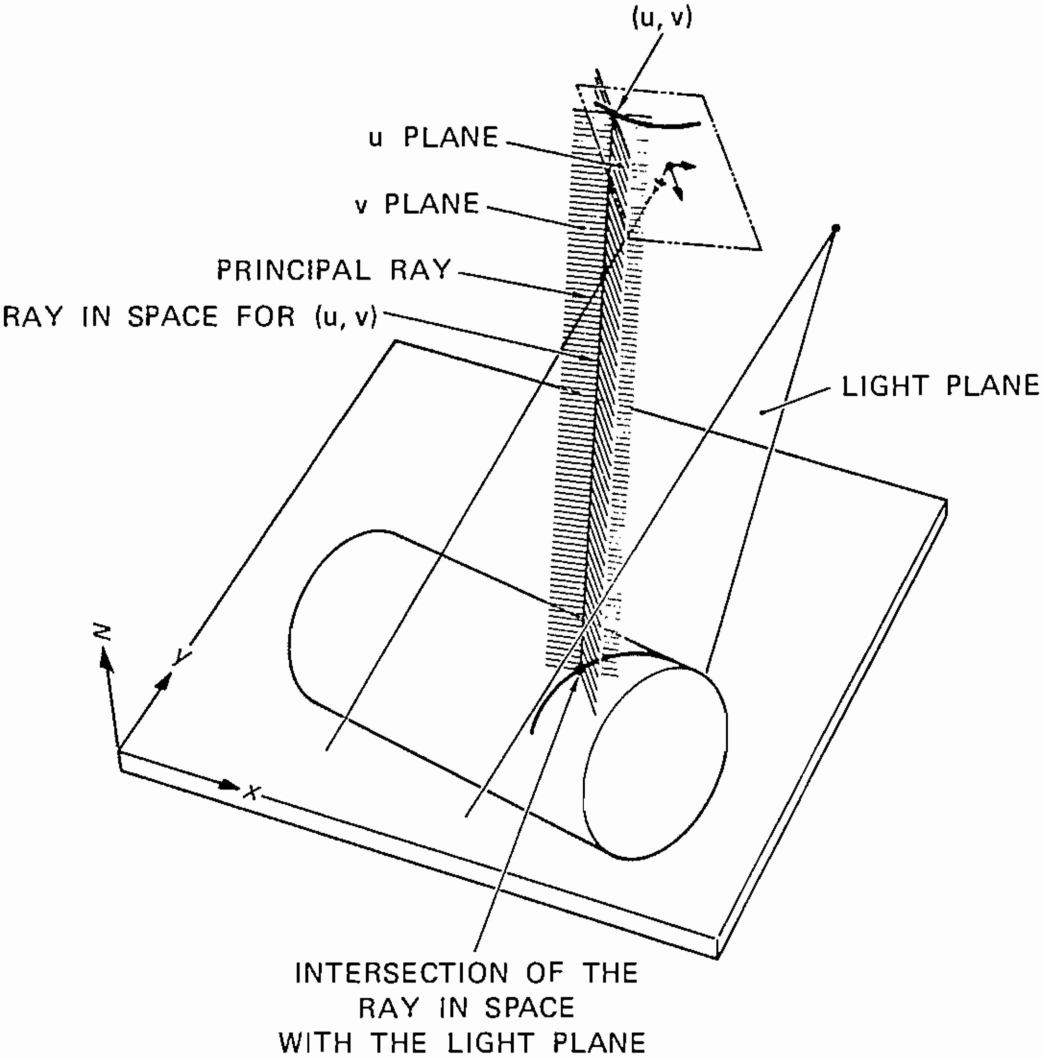


FIGURE 3 AN (x, y, z) POINT DEFINED BY THE INTERSECTION OF THREE PLANES, THE u AND v PLANES, AND THE LIGHT PLANE

$$b1*x + b2*y + b3*z + b4 = 0 \quad , \qquad (6)$$

then the x-y-z position of a point appearing at (u,v) in the image and lying in that plane is determined by the solution to the following three equations:

$$b1*x + \qquad b2*y + \qquad b3*z + \qquad b4 = 0 \quad (7)$$

$$(a11-u*a41)*x + (a12-u*a42)*y + (a13-u*a43)*z + (a14-u*a44) = 0 \quad (8)$$

$$(a21-v*a41)*x + (a22-v*a42)*y + (a23-v*a43)*z + (a24-v*a44) = 0, \quad (9)$$

which can be represented as

$$\begin{pmatrix} b1 & b2 & b3 \\ (a11-u*a41) & (a12-u*a42) & (a13-u*a43) \\ (a21-v*a41) & (a22-v*a42) & (a23-v*a43) \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -b4 \\ (u*a44-a14) \\ (v*a44-a24) \end{pmatrix} (10) \; .$$

Therefore, (x,y,z) can be computed by inverting the 3 X 3 matrix and multiplying the vector on the right side of the equal sign by the inverse.

This solution for (x,y,z) can be applied to some interesting special cases. For example, if one of the coordinates of a point in the world is known, that constraint can be represented as a plane perpendicular to one of coordinate axes and Equation (10) can be used to solve for the other two coordinates of the point. In Figure 2, since the top of the block is at z = 5.0, Equation (10) can be used to compute the x-y-z positions for any point on the top.

Inverting a 3 X 3 matrix for each point to be computed is computationally expensive. Fortunately, it is possible to symbolically invert the matrix in Equation (10) and combine it with the vector on the right so that (x,y,z) can be computed from (u,v) by simply performing the following matrix multiplication:

$$\begin{pmatrix} s*x \\ s*y \\ s*z \\ s \end{pmatrix} = \begin{pmatrix} m11 & m12 & m13 \\ m21 & m22 & m23 \\ m31 & m32 & m33 \\ m41 & m42 & m43 \end{pmatrix} * \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{11}$$

and the three divisions required by the homogeneous representation. Therefore, in this form there are only eight multiplications, eight additions, and three divisions required to compute each x-y-z position.

In this paper we describe how to compute a matrix of mij's as in Equation (11) for a simple range sensor that uses a projected plane of light as the known plane in the world. We refer to the matrix of mij's as the "sensor" matrix, since it depends both on the position of the camera and the position of the light plane.

The intersection of the light plane and the objects appears as a broken line in the camera's image (e.g., see Figure 1(b)). Since the camera is not in the plane of light, the position of the intersection in the image is a function of the height of the object. For the arrangement in Figure 1, the higher the object, the farther right is the intersection. The analysis of an image, such as the one shown in Figure 1(b), produces a sequence of x-y-z positions for points along one slice through the scene. More information can be gathered by moving the objects in front of the sensor. It is also possible to mount the light source and camera on an arm and measure positions relative to the arm. This latter approach has been used at SRI [10], the National Bureau of Standards [11], and elsewhere.

In the remainder of this paper we describe a straightforward procedure for calibrating the sensor, discuss some practical considerations in using it, and conclude with some ideas for extending it to create a fast, multislice range sensor.

II SENSOR CALIBRATION

The calibration of the range sensor is performed in three steps:

* Calibration of the camera.
* Calibration of the light plane.
* Formation of the sensor matrix.

The calibration process is essentially the same, whether the sensor is fixed or mounted on an arm.

A. Calibration of the Camera

Calibrating a camera generally involves determining its characteristic parameters, such as its position, orientation, and focal length (see Figure 2). These parameters can be measured directly or computed from the observed positions of known objects (e.g., see [14-16]). However, for the range sensor and many other applications the actual distances and angles are not important; only the ability to map world points into image points is required. A third method, therefore, which is often easier to perform, is sufficient. In this method the camera matrix is computed directly from the image positions corresponding to known world points.

One way to implement this method is to move a special calibration target to a set of known positions in the camera's field of view and then use the observed positions, together with their world positions, to form a set of linear equations involving the unknown camera matrix elements (i.e., the aij's). These equations can be derived directly from Equations (2) and (3); however, their solution is simplified if the form of the camera matrix (in Equation (1)) is modified slightly. In particular, since w is not used, it is evident that the third row in the matrix is unnecessary. Furthermore, since homogeneous coordinates explicitly include a scale factor, any constant multiple of this matrix represents the same transformation. Therefore, without loss of generality, one of the remaining twelve elements can be arbitrarily set to 1.0. As a result, the task of computing the matrix is reduced to computing the eleven aij's in the following equation:

$$\begin{pmatrix} s*u \\ s*v \\ s*w \\ s \end{pmatrix} = \begin{pmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ 0 & 0 & 1 & 0 \\ a41 & a42 & a43 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} . \qquad (12)$$

After this simplification, the linear equations, derived from Equations (2) and (3) and involving the known world coordinates, their corresponding image locations, and the camera matrix elements, take the form

$$x*a11 + y*a12 + z*a13 + a14 - u*x*a41 - u*y*a42 - u*z*a43 = u \qquad (13)$$

$$x*a21 + y*a22 + z*a23 + a24 - v*x*a41 - v*y*a42 - v*z*a43 = v \quad .(14)$$

Each world-to-image correspondence implies two constraints on the eleven aij's. Sets of these equations can be solved by applying the procedure described in Appendix B to solve an overconstrained set of linear equations. The set of 2*M constraints implied by a set of M world-to-image correspondences can be represented in the following matrix equation:

$$\begin{pmatrix} x1 & y1 & z1 & 1 & 0 & 0 & 0 & 0 & -u1*x1 & -u1*y1 & -u1*z1 \\ 0 & 0 & 0 & 0 & x1 & y1 & z1 & 1 & -v1*x1 & -v1*y1 & -v1*z1 \\ & & & & & & & \cdot & & & \\ & & & & & & & \cdot & & & \\ & & & & & & & \cdot & & & \\ xM & yM & zM & 1 & 0 & 0 & 0 & 0 & -uM*xM & -uM*yM & -uM*zM \\ 0 & 0 & 0 & 0 & xM & yM & zM & 1 & -vM*xM & -vM*yM & -vM*zM \end{pmatrix} * \begin{pmatrix} a11 \\ a12 \\ a13 \\ a14 \\ a21 \\ \cdot \\ \cdot \\ \cdot \\ a43 \end{pmatrix} = \begin{pmatrix} u1 \\ v1 \\ \cdot \\ \cdot \\ \cdot \\ uM \\ vM \end{pmatrix} (15) \quad .$$

The best least-squares estimates for the aij's can be computed from Equation (15) by computing the pseudoinverse of the 2M X 11 matrix and multiplying it by the column vector on the right (see Appendix B). Creating this 2M X 11 matrix and solving for the aij's is precisely the procedure we use at SRI.

Having computed the aij's for the camera matrix, the camera calibration is complete. Equations (2) and (3) can be used to compute the image points corresponding to world points, Equations (4) and (5) define the ray in space corresponding to an image point, and Equation (10) can be used to compute the position of a world point, given an image position and a planar constraint in the world.

This method for calibrating a camera is easy to apply and produces a precise mapping from world to image coordinates, which is exactly what the range finder needs. However, if the values of the individual camera parameters are required (as in visual navigation) a different calibration procedure is probably more appropriate, because it is often difficult to decompose a camera matrix into a set of parameter values.

B. Calibration of the Light Plane

Calibrating the light plane entails computing its equation, which can be done by locating several points on it and applying the plane fitting procedure described in Appendix B. Equation (10), which is based on the camera matrix, can be used to compute the world positions of points in the light plane. For example, if the world coordinate system is defined with respect to the table and a block of known thickness is placed on the table, the top of the block then defines a plane that can be used to determine the positions of points along the intersection of the light plane and the top of the block. By the use of blocks of different thicknesses, points at different heights can be located to define the plane of light. If the sensor is mounted on an arm, the arm can be positioned at different distances from a plane in order to measure a set of points in the light plane.

C. Formation of the Sensor Matrix

Given the camera matrix in Equation (12) and the equation of the light plane in (6), the formation of the sensor matrix simply involves evaluating the following expressions for the mij's and filling in the matrix:

$$m11=(b4*a22-b2*a24)*a43 + (b3*a24-b4*a23)*a42 + (b2*a23-b3*a22) \qquad (16)$$

$$m12=(b2*a14-b4*a12)*a43 + (b4*a13-b3*a14)*a42 + (b3*a12-b2*a13) \qquad (17)$$

$$m13=(b2*a13-b3*a12)*a24 + (b4*a12-b2*a14)*a23 + (b3*a14-b4*a13)*a22 \qquad (18)$$

$$m21=(b1*a24-b4*a21)*a43 + (b4*a23-b3*a24)*a41 + (b3*a21-b1*a23) \qquad (19)$$

$$m22=(b4*a11-b1*a14)*a43 + (b3*a14-b4*a13)*a41 + (b1*a13-b3*a11) \qquad (20)$$

$$m23=(b3*a11-b1*a13)*a24 + (b1*a14-b4*a11)*a23 + (b4*a13-b3*a14)*a21 \qquad (21)$$

$$m31=(b4*a21-b1*a24)*a42 + (b2*a24-b4*a22)*a41 + (b1*a22-b2*a21) \qquad (22)$$

$$m32=(b1*a14-b4*a11)*a42 + (b4*a12-b2*a14)*a41 + (b2*a11-b1*a12) \qquad (22)$$

$$m33=(b1*a12-b2*a11)*a24 + (b4*a11-b1*a14)*a22 + (b2*a14-b4*a12)*a21 \qquad (24)$$

$$m41=(b2*a21-b1*a22)*a43 + (b1*a23-b3*a21)*a42 + (b3*a22-b2*a23)*a41 \qquad (25)$$

$$m42=(b1*a12-b2*a11)*a43 + (b3*a11-b1*a13)*a42 + (b2*a13-b3*a12)*a41 \qquad (26)$$

$$m43=(b2*a11-b1*a12)*a23 + (b1*a13-b3*a11)*a22 + (b3*a12-b2*a13)*a21.(27)$$

This completes the calibration of the sensor. Given the sensor matrix, Equation (11) can be used to compute the three-dimensional positions of points along the intersection of the light plane and objects in the scene.

The MACSYMA system at MIT [17] was used to invert the matrix in Equation (10) symbolically and form the expressions for the mij's. Agin was the first to indicate that it was possible to combine a camera matrix and an equation of a plane to form this type of sensor matrix. He gave a different derivation of the sensor matrix in [18].

III PRACTICAL CONSIDERATIONS

Figure 4 shows one implementation of this type of sensor that we have used at SRI. Objects are mounted on an x-y table so they can be moved underneath the sensor. The camera is a General Electric TN2500 with a spatial resolution of 240 X 240 pixels. The light source, which is not shown in the photograph, is a laser whose beam is spread into a plane by a cylindrical lens. It is not necessary to use a laser; indeed, we have also used a standard 35-mm slide projector with a slide that is completely black except for one thin line. (It is easy to make such a slide by taking a black-and-white negative picture of a thin line and mounting the negative in a slide holder.)

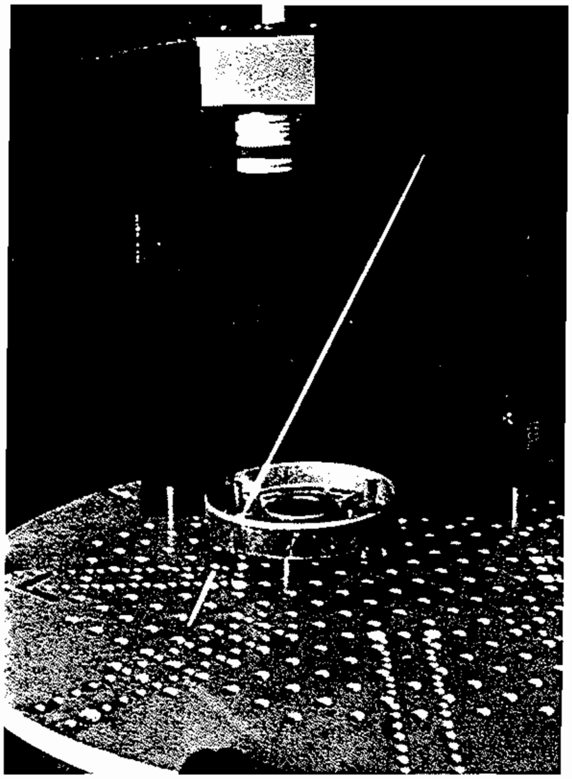


FIGURE 4 AN SRI RANGE SENSOR

One problem with range systems based on triangulation, as this one is, is that there are "shadow" areas caused by portions of the scene that are not "visible" to both the camera and projector. The farther apart these are, the larger the shadow areas. Shadow areas can be minimized by placing the camera and projector close to each other--but, the closer they are, the coarser the height resolution will be. Height resolution can be maintained by increasing the focal length of the camera lens and reducing the field of view. Thus, there are trade-offs between shadow area, height resolution, and field of view. The light plane in Figure 4 is approximately sixty degrees from horizontal, which we've found to be a reasonable compromise.

We calibrate the camera in this device by inserting posts of known heights in a hole in the table, moving the table to predetermined positions, measuring the image locations of the center of the top of the post, and computing the camera matrix from these world-to-image pairs (by creating and solving Equation (15)). We calibrate the light plane by setting blocks of known thicknesses on the table, measuring the three-dimensional positions of points in the plane (through the use of the camera matrix and Equation (10)), and then fitting a plane to the measured points.

If the world points used to compute the camera matrix contain a large constant offset (e.g., if they are distributed about (-37,48,30)), the numerical precision of the camera calibration can be improved by subtracting the offset from the world coordinates before computing the camera matrix, and then correcting the matrix for this displacement. The corrected matrix of cij's is formed as follows:

$$\begin{pmatrix} c11 & c12 & c13 & c14 \\ c21 & c22 & c23 & c24 \\ c31 & c32 & c33 & c34 \\ c41 & c42 & c43 & c44 \end{pmatrix} = \begin{pmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ 0 & 0 & 1 & 0 \\ a41 & a42 & a43 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & -Dx \\ 0 & 1 & 0 & -Dy \\ 0 & 0 & 1 & -Dz \\ 0 & 0 & 0 & 1 \end{pmatrix} , \qquad (28)$$

where the matrix of aij's was computed from the adjusted (x,y,z)'s and (Dx,Dy,Dz) is the offset.

To check the camera calibration, we use the derived camera matrix to map the post positions into the image and then compare the observed image positions with these predictions. The errors are generally less than half a pixel. We check the equation of the light plane by measuring the distances of the points from the fitted plane. When the field of view of our 240 X 240 camera is a ten-inch square and the angle of the light plane is sixty degrees from horizontal, a horizontal shift in the image of one pixel corresponds to a change in z of approximately .08 of an inch. The relative precision of this setup is quite good, generally less than .01 of an inch. The absolute precision is generally less than the error associated with one pixel, which is .08 of an inch.

Figure 5 is a composition of 50 individually processed slices from a pair of cylindrical castings, one lying on top of the other. Figure 6 shows examples of some of the steps entailed in processing an a single slice. The intersection of the plane of light and an object is a relatively thin line but, since there is some thickness, it appears in an image as a line several pixels wide. Which points in the image should be defined to be on the plane? For simplicity, most range acquisition systems that use a projected light plane have selected one point in each row of the image (e.g., see [7]). Usually they choose the middle pixel of the first long run of pixels that are "on." However, since the images are perspective images, there may be more than one valid intersection per row. More importantly, when the intersection line is horizontal in the image, the row centers are completely wrong (see Figure 7). Since important data can be obtained from horizontal or almost horizontal intersections, we apply a thinning algorithm (see [19]) to produce a center line that is independent of the intersection's orientation in the image. This centerline still may not lie in the light plane, but it is close--and to a first order it is independent of the camera's focus and the threshold used to produce the binary picture.

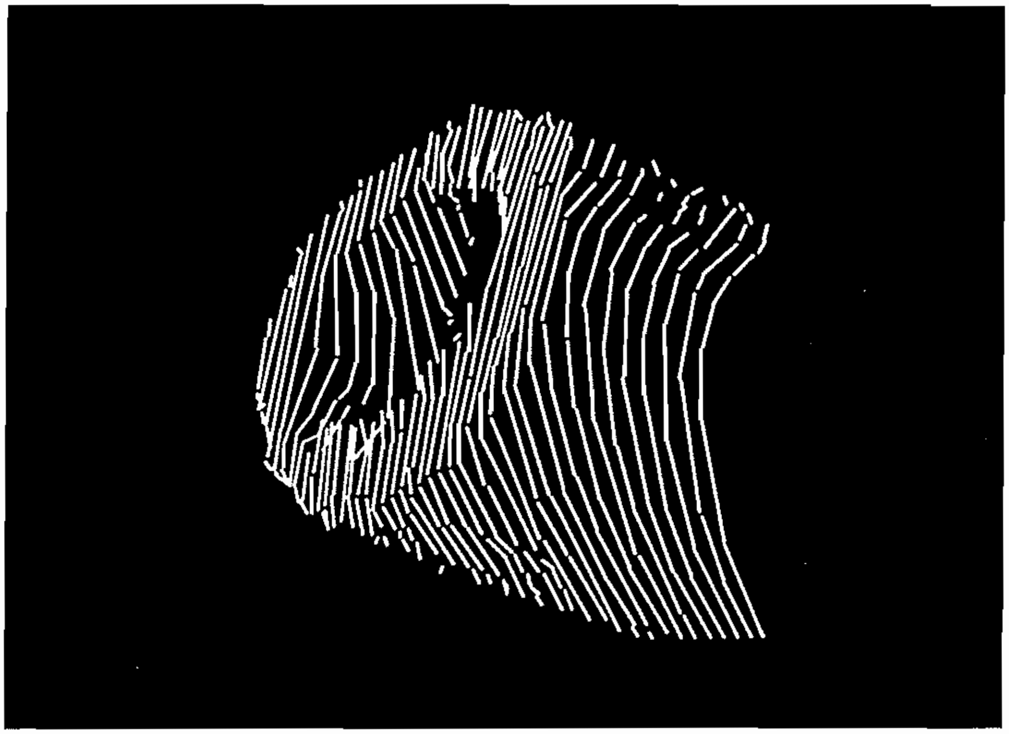
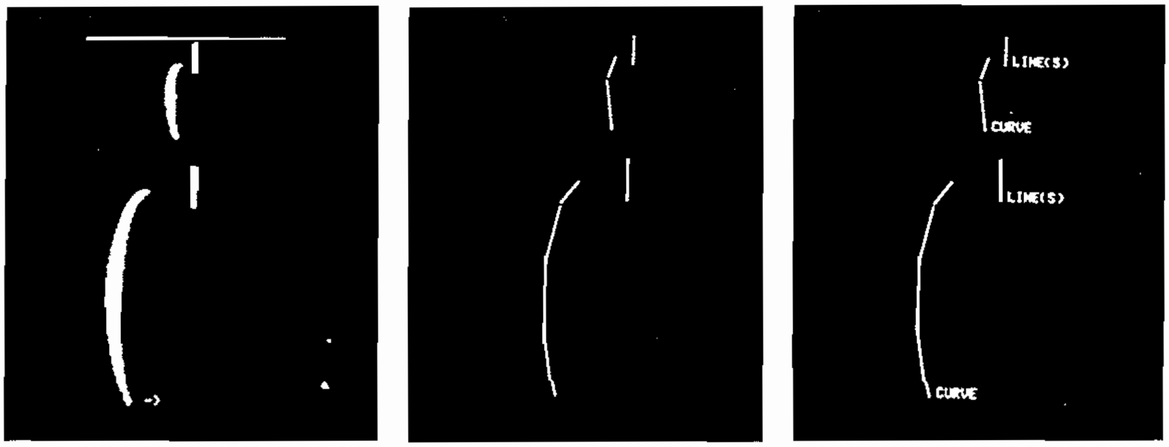


FIGURE 5 A PICTURE CONSTRUCTED FROM 50 SEPARATE SLICES TAKEN AT THE END OF A CYLINDRICAL CASTING



(a) THE THRESHOLDED IMAGE (b) THE FILTERED DATA (c) THE CLASSIFIED SEGMENTS

FIGURE 6 RESULTS OF THREE STEPS IN THE PROCESSING OF AN INTERSECTION

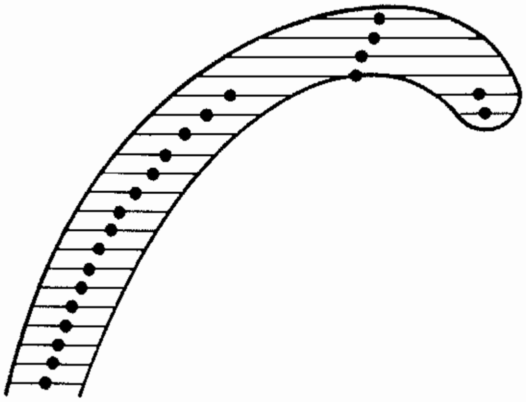


FIGURE 7 EXAMPLE OF THE PROBLEM WITH ROW-BY-ROW CENTERS

IV CONCLUSION

The range sensor described in this paper is simple to set up, relatively easy to calibrate, and quite precise. However, it only measures the positions of points along one slice through a scene. There are several ways to extend this device to measure a region of the scene, such as moving the scene in front of the sensor or scanning the light plane over the scene. One of the most promising ideas, suggested by Altschuler et al., is to project a sequence of coded patterns of planes onto the scene and use the pattern of occurrences of an intersection point to identify the plane producing it, which in turn specifies the sensor matrix that converts the point into its three-dimensional coordinates [20,21]. Ideas such as this make us optimistic that inexpensive, high-resolution range sensors will be available in the not-too-distant future.

Appendix A

HOMOGENEOUS COORDINATES

Appendix A

HOMOGENEOUS COORDINATES

In homogeneous coordinates a three-dimensional point, (x,y,z), is represented as a 4-vector, (x′,y′,z′,s), where s is a scale factor such that

$$x = \frac{x'}{s},\ y = \frac{y'}{s},\ z = \frac{z'}{s} \quad . \qquad \text{(A1)}$$

This representation makes it possible to represent a camera′s perspective transformation, which is not linear with respect to three-dimensional vectors (x,y,z), as a linear operation on 4-vectors. The advantage of representing transformations as linear operations is that there is a well-developed theory of linear algebra that can be applied.

To get some insight into how homogeneous coordinates work, consider Figure A-1, which is a diagram of a one-dimensional pinhole camera. The camera has a focal length of F, which is assumed to be known. By similar triangles, the image position u, of a point at a distance z from the lens center and x off the optical axis, is given by

$$u = \frac{x*F}{z} \quad . \qquad \text{(A2)}$$

For a two-dimensional camera there is a similar expression for the second coordinate of the image position:

$$v = \frac{y*F}{z} \quad . \qquad \text{(A3)}$$

Since u and v are not linear functions of the variables x, y, and z, it is not possible to represent them in a matrix equation such as

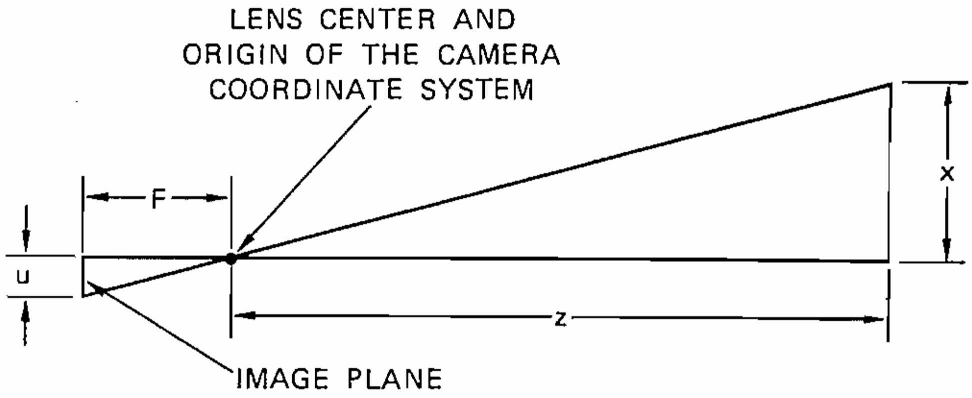


FIGURE A-1 GEOMETRY OF A ONE-DIMENSIONAL PINHOLE CAMERA

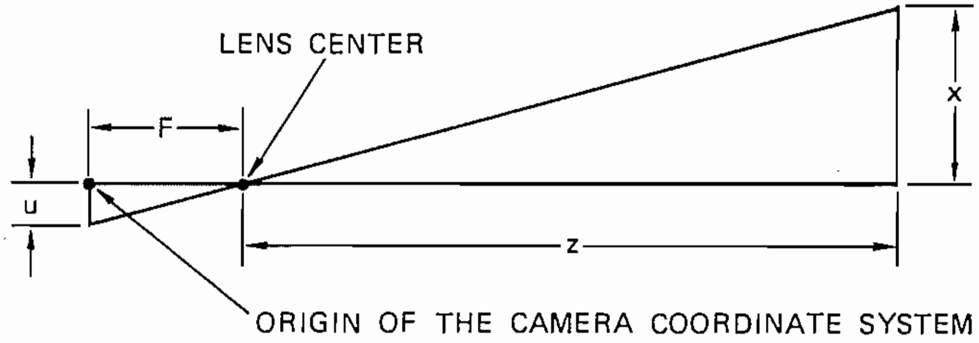


FIGURE A-2 A REVISED MODEL OF A ONE-DIMENSIONAL PINHOLE CAMERA

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} . \qquad (A4)$$

However, if homogeneous coordinates are used, the perspective transform can be represented as follows:

$$\begin{pmatrix} s*u \\ s*v \\ s*w \\ s \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/F & 0 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} . \qquad (A5)$$

Performing the multiplication on the right side of this equation produces

$$\begin{pmatrix} s*u \\ s*v \\ s*w \\ s \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ z/F \end{pmatrix} , \qquad (A6)$$

which implies the desired expressions for u and v shown in Equations (A2) and (A3). The variable w in this formulation equals to F for all values of (x,y,z) and can be ignored.

In practice, a slightly different model of the camera is generally used that leads to a perspective matrix that is invertible, unlike the one in Equation (A4). In the new model (see Figure A-2) the origin of the camera coordinate system is defined to be in the image plane instead of at the lens center. This shift of the origin leads to slightly different expressions for u and v:

$$u = \frac{x * F}{(z + F)} \qquad v = \frac{y * F}{(z + F)} , \qquad (A7)$$

which in turn lead to a slightly different matrix for the perspective transformation:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/F & 1 \end{pmatrix} . \qquad \text{(A8)}$$

A transformation representing an arbitrarily positioned camera (that projects three-dimensional "world" points onto a two-dimensional image plane) involves displacements, rotations, and scale changes in addition to a perspective transformation. All of these operations are linear and can be represented as 4 X 4 matrices that map one set of homogeneous coordinates into another. For example, a rotation of theta about the principal axis of the camera is

$$\begin{pmatrix} \cos(\text{theta}) & \sin(\text{theta}) & 0 & 0 \\ -\sin(\text{theta}) & \cos(\text{theta}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \text{(A9)}$$

and a (Dx,Dy,Dz) displacement of the camera from the origin of the world coordinate system is

$$\begin{pmatrix} 1 & 0 & 0 & -Dx \\ 0 & 1 & 0 & -Dy \\ 0 & 0 & 1 & -Dz \\ 0 & 0 & 0 & 1 \end{pmatrix} . \qquad \text{(A10)}$$

Since all of the transformations are linear, they can be multiplied together to form more complex linear transformations.

A typical camera matrix is a composition of several transformations performed in a specific order, such as

(1) A displacement from the origin

(2) A rotation about the z axis (i.e., heading)

(3) A rotation about the new y axis (i.e., pitch)

(4) A rotation about the new x axis (i.e., roll)

(5) A perspective transformation

(6) A displacement from the center of the image

(7) A scale change.

In this paper the specific choice of parameters and their assumed order of operation are not important. Only the form of the final matrix that can represent an arbitrarily positioned and oriented camera is important, and that can be characterized by the matrix in the following equation:

$$\begin{pmatrix} s*u \\ s*v \\ s*w \\ s \end{pmatrix} = \begin{pmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \\ a41 & a42 & a43 & a44 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} . \qquad \text{(A11)}$$

A more complete description of homogeneous coordinates and their use in camera modeling can be found in [22, 23].

Appendix B

OVERCONSTRAINED SETS OF LINEAR EQUATIONS

Appendix B

OVERCONSTRAINED SETS OF LINEAR EQUATIONS

Many model fitting tasks can be stated as a set of M linear equations to be solved for N unknown parameters. For example, consider the task of fitting a plane

$$a1*X + a2*Y + a3*Z + a4 = 0 \qquad \text{(B1)}$$

to a set of three-dimensional points (xi,yi,zi), i = 1 to M. As stated the task is to compute the four coefficients a1 through a4. However, the trivial solution in which all of the ai's are zero satisfies any number of these equations. This possibility can be eliminated by reformulating Equation (B1). Dividing (B1) by a3 and relabeling produces the following equation for the plane

$$b1*X + b2*Y + Z + b3 = 0 \quad , \qquad \text{(B2)}$$

which can be rewritten as

$$X*b1 + Y*b2 + b3 = -Z \quad . \qquad \text{(B3)}$$

This form of the equation cannot represent a plane perpendicular to the z axis (i.e., parallel to the x-y plane). However, if the plane is known not to be perpendicular to the z axis, Equation (B3) is a convenient form for fitting, because it contains three linearly related unknowns (b1, b2, and b3) and a constant term (-Z).

Given M points that are supposed to be on a plane, each of them contributes one constraint in the form of Equation (B3). The M equations can be combined into one matrix equation

$$\begin{pmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ xM & yM & 1 \end{pmatrix} * \begin{pmatrix} b1 \\ b2 \\ b3 \end{pmatrix} = \begin{pmatrix} -z1 \\ -z2 \\ \cdot \\ \cdot \\ -zM \end{pmatrix} , \qquad (B4)$$

which can be rewritten as

$$A * B = C \quad . \qquad (B5)$$

If the data to be fitted contain fewer than three points, the solution is underconstrained and Equation (B5) cannot be solved. If the data contain exactly three points, A is a square matrix and the "exact" solution is

$$B = A^{-1} * C \quad , \qquad (B6)$$

assuming the inverse of A (represented as A^{-1}) exists, which will be true as long as the three points are not colinear.

If the data contain more than three points, the solution is overconstrained and Equation (B5) cannot be solved directly, because A is rectangular and does not have an inverse. The problem, then, is to compute the plane that fits the data as well as possible. If the measure of goodness of fit is taken to be the sum of the squares of the distances of the points from the plane, the solution is

$$B = (A^T * A)^{-1} * A^T * C \quad , \qquad (B7)$$

where A^T represents the transpose of A [24]. $(A^T * A)^{-1} * A^T$ is called the pseudoinverse of A. Notice that $(A^T * A)$ is always a square matrix. For example, given twenty points that are supposed to be on a plane, A is a 20 X 3 matrix, A^T is a 3 X 20 matrix, and $(A^T * A)$ is a 3 X 3 matrix, which can be inverted.

B having been computed, the equation of the fitted plane is

$$b1*X + b2*Y + Z + b3 = 0 \quad , \qquad (B8)$$

which is the best least-squares fit of a plane to the data.

This fitting technique generalizes to any number of unknowns. For example, it can be used to compute the second-order equation of two variables that fits a set of two-dimensional points. Consider the equation

$$a1*X^2 + a2*X*Y + a3*Y^2 + a4*X + a5*Y + a6 = 0 \quad . \qquad (B9)$$

Dividing (B9) by a1, relabeling, and transferring the "constant" term to the right produces

$$X*Y*b1 + Y^2 *b2 + X*b3 + Y*b4 + b5 = -X^2 \quad , \qquad (B10)$$

which contains five unknowns (b1 through b5). Given M points, where M is greater than or equal to 5, the matrices A and C can be formed as they were in the plane-fitting example and the best least-squares solution for the curve's parameters (i.e., the bi's) is given by Equation (B7).

This technique is a straightforward way to compute the best least-squares set of linearly related parameters, but one has to be careful to make sure that the linear equation represents the desired class of functions. In the second-order-equation fitting example described above, there are some parabolas and hyperbolas that the particular form of the equation cannot represent. If that is not satisfactory, a different form should be derived. If the class of functions to be fitted is inherently nonlinear in the parameters to be computed, this technique cannot be applied directly. However, it can be imbedded in an iterative procedure to solve a nonlinear problem (e.g., see [25]).

REFERENCES

1. M. J. Hannah, "Computer Matching of Areas in Stereo Images," Ph.D. thesis, Stanford Artificial Intelligence Project Memo AIM-239, Stanford University, Stanford, California (July 1974).

2 D. Marr and T. Poggio, "Cooperative Computation of Stereo Disparity," Science, Vol. 194, pp. 283-287 (October 1976).

3. B. K. P. Horn, "Obtaining Shape from Shading Information," The Psychology of Computer Vision, P. H. Winston, ed., (McGraw-Hill Book Company, New York, New York, 1975).

4. B. K. P. Horn, "Understanding Image Intensities," Artificial Intelligence, Vol. 8, No. 2, pp. 201-231 (Spring 1977).

5. D. Marr, "Representing Visual Information," Computer Vision Systems, A. R. Hanson and E. M. Riseman, eds., pp. 61-80 (Academic Press, New York, New York, 1978).

6. H. G. Barrow and J. M. Tenenbaum, "Recovering Intrinsic Scene Characteristics from Images," Computer Vision Systems, A. R. Hanson and E. M. Riseman, eds., pp. 3-26 (Academic Press, New York, New York, 1978).

7. R. J. Popplestone, et al., "Forming Models of Plane-and-Cylinder Faceted Bodies from Light Stripes," Proceedings of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, pp. 664-668 (August 1975).

8. G. J. Agin and T. O. Binford, "Computer Description of Curved Objects," Proceedings of the Third International Joint Conference on Artificial Intelligence, Stanford University, Stanford, California, pp. 629-640 (August 1973).

9. Y. Shirai and M. Suwa, "Recognition of Polyhedrons with a Range Sensor," Proceedings of the Second International Joint Conference on Artificial Intelligence, London, England, pp. 80-87 (September 1971).

10. G. J. Agin, "Real-Time Control of a Robot with a Mobile Camera," Proceedings Ninth International Symposium on Industrial Robots, Washington, D. C., pp. 233-246 (March 1979).

11. G. J. VanderBrug, J. S. Albus, and E. Barkmeyer, "A Vision System for Real-Time Control of Robots," *Proceedings Ninth International Symposium on Industrial Robots*, Washington, D. C., pp. 213-231 (March 1979).

12. D. Nitzan, et al., "Machine Intelligence Research Applied to Industrial Automation," Tenth Report, NSF Grant DAR78-27128, SRI International, Menlo Park, California (November 1980).

13. R. C. Bolles and M. A. Fischler, "A RANSAC-Based Approach to Model-Fitting and its Application to Finding Cylinders in Range Data," to appear in the *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia, Canada (August 1981).

14. I. Sobel, "On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes," *Artificial Intelligence*, Vol. 5, No. 2, pp. 185-198 (Summer 1974).

15. D. B. Gennery, "Least-Squares Stereo-Camera Calibration," Stanford Artificial Intelligence Project internal memo, Stanford University, Stanford, California (1975).

16. A. M. Thompson, "Camera Geometry for Robot Vision," *Robotics Age*, pp. 20-27 (March 1981).

17. The Mathlab Group, "MACSYMA Reference Manual," Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts (December 1977).

18. G. J. Agin, "Collineation Between a Plane of Light and an Image," Personal Notes (January 1980).

19. C. J. Hilditch, "Linear Skeletons from Square Cupboards," *Machine Intelligence* Vol. IV, pp. 403-420 (1969).

20. B. R. Altschuler, M. D. Altschuler, and J. Toboada, "Measuring Surfaces Space-Coded by a Laser-Projected Dot Matrix," *Proceedings of the SPIE Technical Symposium on Imaging Applications for Automated Industrial Inspection and Assembly*, Washington, D. C. (April 1979).

21. B. R. Altschuler, J. Toboada, and M. D. Altschuler, "A Laser Electro-Optical System for 3-D Topographic Mensuration," *Proceedings of the SPIE Technical Symposium on Imaging Applications for Automated Industrial Inspection and Assembly*, Washington, D. C. (April 1979).

22. R. F. Sproull and W. M. Newman, *Principles of Interactive Computer Graphics* (McGraw Hill Book Company, New York, New York, 1973).

23. R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis (Wiley-Interscience, New York, New York, 1973).

24. F. A. Graybill, An Introduction to Linear Statistical Models, Volume I, (McGraw-Hill Book Company, New York, New York 1961).

25. R. C. Bolles, "Verification Vision Within a Programmable Assembly System," Ph.D. thesis, Stanford Artificial Intelligence Project Memo AIM-295, Stanford University, Stanford, California (December 1976).