SAND96-2346C

CONF-970469--2

Rapid World Modeling:
Fitting Range Data to Geometric Primitives

John Feddema and Charles Little[1]

Sandia National Laboratories
Albuquerque, NM 87185

MASTER

RECEIVED
OCT 03 1996
OSTI

Abstract

World modeling is defined as the process of creating a numerical geometric model of a real world environment or workspace. This model is often used in robotics to plan robot motions which perform a task while avoiding obstacles. In many applications where the world model does not exist ahead of time, structured lighting, laser range finders, and even acoustical sensors have been used to create three dimensional maps of the environment. These maps consist of thousands of range points which are difficult to handle and interpret. This paper presents a least squares technique for fitting range data to planar and quadric surfaces, including cylinders and ellipsoids. Once fit to these primitive surfaces, the amount of data associated with a surface is greatly reduced up to three orders of magnitude, thus allowing for more rapid handling and analysis of world data.

1. Introduction

For the past seven years, Sandia National Laboratories has been active in the development of robotic systems to help remediate DOE's waste sites and decommissioned facilities. Some of these facilities have high levels of radioactivity which prevent manual clean-up. Tele-operated and autonomous robotic systems have been envisioned as the only suitable means of removing the radioactive elements.

Early prototype systems have demonstrated the feasibility of using a computer generated graphical interface to navigate and control these systems. Much like a video game, this graphical environment contains a 3 dimensional (3D) map, or world model, of the facility. The traditional method of generating the model has been to use 3D CAD tools. Unfortunately, this is a very manual, time consuming operation, primarily due to the time needed to collect the measurements. Blueprints are often unavailable and rarely match the as-built conditions. Changes, modifications, or simply adding or moving equipment around add many hours to construct the model. The problem is increased when data cannot be gathered because access to the facility is limited due to environmental hazards. One solution is to use remotely deployed range sensors. These devices are used to scan over an area, and provide 3D surface information that can be used to create a surface map [1].

These sensors are capable of generating extremely dense range data which when converted directly to a world model may contain hundreds of thousands of polygons [2]. The world model created from many scans is so large that even the newest, most expensive computer graphics workstations have a difficult time displaying this data in real time. In addition, this type of brute force representation of objects makes it impossible to perform automated path planning or disassembly because of the exorbitant computational cost of considering each scanned point on the surface.

What is needed is a way to reduce the data sets to a finite number of primitives. Luckily, the facilities that we are interested in are mostly man-

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.

made, and a significant number of man-made objects can be segmented into primitives such as planes, cylinders, and spheres. For example, walls, cabinets, tables, barrels, pipes, and hemispherical domes can be fit with these primitives.

In the past, researchers have investigated fitting range data using free-form surface models such as B-splines, NURBS, or Dual Kriging [3][4]. These techniques fit parametric equations, often low order polynomials, to scattered data points within a rectangular patch. The control points of each patch are defined such that smooth surfaces are generated between adjacent patches over an entire surface. More recently, Garcia [5][6] has developed triangular meshes which incorporate uncertainty into the control points through shape modifiers. The goal of this paper is not to create smooth surfaces that precisely model the surface, but to create simple primitives that approximate the real data and are easy to represent in a world model. Similar work in finding planar and quadric surfaces for inclusion in a world model is described in [7]. In [7], the object is recognized and its position and orientation determined by comparing model surface patches from a user selected shape to scene surface patches. In this work, no pre-defined user selected surfaces are needed.

This paper describes how to fit large range data sets to planes, cylinders, and spheres using a least squares approach. Section 2 describes the least squares fit routines for planar surfaces, and Section 3 describes the least squares fit for quadric surfaces. Section 4 presents experimental results, and Section 5 summarizes the results and concludes with suggested future research and development.

2. Planar Fit

This section describes how to fit dense range map data to a planar surface using a least squares technique. It is assumed that the range data has already been segmented to isolate points on the plane from non-planar points. Currently, this is performed manually in a 3D graphical environment developed in AVS from Advanced Visual Systems, Incorporated. Once segmented, the points are sent to a routine which uses the following least squares technique to fit the points

to an implicit representation of a plane. From this implicit representation, the position and extent of the plane are determined.

The implicit equation of a plane is

$$f(x,y,z) = n_x x + n_y y + n_z z + 1 = 0 \qquad (1)$$

where n_x, n_y, and n_z are constants which represent the plane's normal vector. The least squares solution minimizes the expression

$$S = \sum_{i=1}^{n} \left[f(x_i, y_i, z_i) \right]^2. \qquad (2)$$

Taking the partial of S with respect to n_x, n_y, and n_z and setting the partials equal to zero, the least squares solution is

$$\mathbf{n} = \mathbf{A}^{-1} \mathbf{b} \qquad (3)$$

where

$$\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i z_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i z_i \\ \sum x_i z_i & \sum y_i z_i & \sum z_i^2 \end{bmatrix},$$

$$\text{and } \mathbf{b} = \begin{bmatrix} -\sum x_i \\ -\sum y_i \\ -\sum z_i \end{bmatrix}. \qquad (4)$$

From Equations (3) and (4), a data set which might contain several thousand points can be reduced to three parameters n_x, n_y, and n_z. In addition to these three parameters, we also want to determine the extent of the plane. One way of doing this is with the following algorithm.

1. Project all points onto the planar surface determined by Equations (3) and (4). The projection of any point $\vec{r}_i$ onto the plane is given by

$$\left(\vec{r}_p \right)_i = \vec{r}_i + \left(p - \vec{r}_i \bullet \vec{u} \right) \vec{u} \qquad (5)$$

where the perpendicular distance from the plane to the origin is

2

$$p = \frac{1}{\sqrt{n_x^2 + n_y^2 + n_z^2}} \qquad (6)$$

and the normal unit vector of the plane is

$$\vec{u} = \frac{-1}{\sqrt{n_x^2 + n_y^2 + n_z^2}} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} . \qquad (7)$$

2. Find the center of the points.

$$\vec{r}_{cg} = \frac{1}{n} \sum_{i=1}^{n} \left(\vec{r}_p\right)_i \qquad (8)$$

3. Find the point which is the farthest distance from the center. Define this maximum point as the x-axis of the plane. The y-axis can be determined from the cross product of the z-axis ($\vec{u}$) and x-axis.

$$\hat{x} = \left\{ \left(\vec{r}_p\right)_i \Big|\, \max_{i=1,\dots,n} \left\| \left(\vec{r}_p\right)_i - \vec{r}_{cg} \right\| \right\} \qquad (9)$$

$$\hat{y} = \frac{\vec{u} \times \hat{x}}{\left\| \vec{u} \times \hat{x} \right\|} \qquad (10)$$

4. Find the points which form the extent of the plane. One way to do this is to imagine a deformable circle in the plane whose m node points around the circle are the planar points that are furthest from the center within a discretized sector of the circle. The angle of each point with respect to the plane's x-axis is given by

$$\left(\theta_p\right)_i = a\tan 2\left(\left(x_p\right)_i , \left(y_p\right)_i \right) \qquad (11)$$

where

$$\left(x_p\right)_i = \left(\left(\vec{r}_p\right)_i - \vec{r}_{cg} \right) \bullet \hat{x} \qquad (12)$$

and

$$\left(y_p\right)_i = \left(\left(\vec{r}_p\right)_i - \vec{r}_{cg} \right) \bullet \hat{y} . \qquad (13)$$

The points that form the extent are given by

$$\left(\vec{r}_e\right)_j = \left\{ \left(\vec{r}_p\right)_i \Bigg| \begin{matrix} \max\limits_{i=1,\dots,n} \left\| \left(\vec{r}_p\right)_i - \vec{r}_{cg} \right\| \wedge \\ \left[\dfrac{\pi(2j-3)}{m} < \left(\theta_p\right)_i \le \dfrac{\pi(2j-1)}{m} \right] \end{matrix} \right\} \qquad (14)$$

for $j = 1,\dots,m$. The operator must select m.

Figure 1 is an example of planar fitting. The spheres show the vertices of the original data set. The surface polygons were created using the circular algorithm presented above. The outer vertices of the surface polygons use the projected planar vertices from the original data set. The common center vertex is derived from the centroid calculation. The advantage of this method is simplicity and therefore speed. With a reasonable number of divisions, it captures edge detail a convex hull would miss, without the complexity of a perimeter search based on a threshold length.
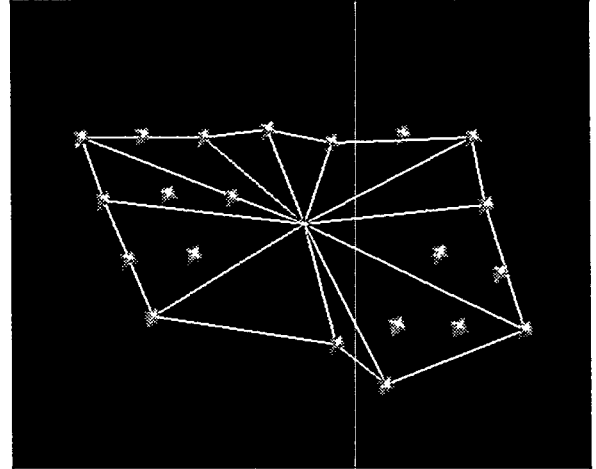

Figure 1. Planar fit.

3. Quadric Fit

This section describes how to fit dense range map data to a quadric surface, which includes cylinders, ellipsoids, cones, and hyperboloids. Once the data is fit to a quadric equation, the parameters of the quadric indicate the type of quadric, e.g., cylinder, ellipsoid, etc. These parameters can also be used to determine the position and orientation of the object as well as variables such as the radius of a cylinder.

The implicit equation of a quadric surface is

$$f(x,y,z) = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a & h & g & u \\ h & b & f & v \\ g & f & c & w \\ u & v & w & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0 \quad (15)$$

where a, b, c, f, g, h, u, v, and w are constants.

The least squares solution again minimizes

$$S = \sum_{i=1}^{n} \left[f(x_i, y_i, z_i) \right]^2. \quad (16)$$

Taking the partial of S with respect to a, b, c, f, g, h, u, v, and w and setting the partials equal to zero, the least squares solution is

$$\theta = A^{-1} b \quad (17)$$

where

$$\theta = \begin{bmatrix} a \\ b \\ c \\ h \\ g \\ f \\ u \\ v \\ w \end{bmatrix}, \quad b = \begin{bmatrix} -\sum x_i^2 \\ -\sum y_i^2 \\ -\sum z_i^2 \\ -\sum x_i y_i \\ -\sum x_i z_i \\ -\sum y_i z_i \\ -\sum x_i \\ -\sum y_i \\ -\sum z_i \end{bmatrix}, \quad \text{and}$$

$$A = \begin{bmatrix} \sum x_i^4 & \sum x_i^2 y_i^2 & \sum x_i^2 z_i^2 & 2\sum x_i^3 y_i & 2\sum x_i^3 z_i & 2\sum x_i^2 y_i z_i & 2\sum x_i^3 & 2\sum x_i^2 y_i & 2\sum x_i^2 z_i \\ \sum x_i^2 y_i^2 & \sum y_i^4 & \sum y_i^2 z_i^2 & 2\sum x_i y_i^3 & 2\sum x_i y_i^2 z_i & 2\sum y_i^3 z_i & 2\sum x_i y_i^2 & 2\sum y_i^3 & 2\sum y_i^2 z_i \\ \sum x_i^2 z_i^2 & \sum y_i^2 z_i^2 & \sum z_i^4 & 2\sum x_i y_i z_i^2 & 2\sum x_i z_i^3 & 2\sum y_i z_i^3 & 2\sum x_i z_i^2 & 2\sum y_i z_i^2 & 2\sum z_i^3 \\ \sum x_i^3 y_i & \sum x_i y_i^3 & \sum x_i y_i z_i^2 & 2\sum x_i^2 y_i^2 & 2\sum x_i^2 y_i z_i & 2\sum x_i y_i^2 z_i & 2\sum x_i^2 y_i & 2\sum x_i y_i^2 & 2\sum x_i y_i z_i \\ \sum x_i^3 z_i & \sum x_i y_i^2 z_i & \sum x_i z_i^3 & 2\sum x_i^2 y_i z_i & 2\sum x_i^2 z_i^2 & 2\sum x_i y_i z_i^2 & 2\sum x_i^2 z_i & 2\sum x_i y_i z_i & 2\sum x_i z_i^2 \\ \sum x_i^2 y_i z_i & \sum y_i^3 z_i & \sum y_i z_i^3 & 2\sum x_i y_i^2 z_i & 2\sum x_i y_i z_i^2 & 2\sum y_i^2 z_i^2 & 2\sum x_i y_i z_i & 2\sum y_i^2 z_i & 2\sum y_i z_i^2 \\ \sum x_i^3 & \sum x_i y_i^2 & \sum x_i z_i^2 & 2\sum x_i^2 y_i & 2\sum x_i^2 z_i & 2\sum x_i y_i z_i & 2\sum x_i^2 & 2\sum x_i y_i & 2\sum x_i z_i \\ \sum x_i^2 y_i & \sum y_i^3 & \sum y_i z_i^2 & 2\sum x_i y_i^2 & 2\sum x_i y_i z_i & 2\sum y_i^2 z_i & 2\sum x_i y_i & 2\sum y_i^2 & 2\sum y_i z_i \\ \sum x_i^2 z_i & \sum y_i^2 z_i & \sum z_i^3 & 2\sum x_i y_i z_i & 2\sum x_i z_i^2 & 2\sum y_i z_i^2 & 2\sum x_i z_i & 2\sum y_i z_i & 2\sum z_i^2 \end{bmatrix} \quad (18)$$

Once the quadric coefficients are known, it is possible to determine the type of quadric from the rank, determinant, and eigenvalues of the following matrices.

$$D = \begin{bmatrix} a & h & g \\ h & b & f \\ g & f & c \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} a & h & g & u \\ h & b & f & v \\ g & f & c & w \\ u & v & w & 1 \end{bmatrix}$$

4

Let $\lambda_1, \lambda_2, \lambda_3$ = eigenvalues($\mathbf{D}$).

The following table is used in this decision.

Table 1. Types of quadrics [8].

Quadric Surface	rank($\mathbf{D}$)	rank($\mathbf{E}$)	sign of det($\mathbf{E}$)	Nonzero $\lambda_1, \lambda_2, \lambda_3$ same sign?
Real ellipsoid	3	4	-	yes
Imaginary ellipsoid	3	4	+	yes
Hyperboloid of one sheet	3	4	+	no
Hyperboloid of two sheets	3	4	-	no
Real quadric cone	3	3		no
Imaginary quadric cone	3	3		yes
Elliptic paraboloid	2	4	-	yes
Hyperbolic paraboloid	2	4	+	no
Real elliptic cylinder	2	3		yes
Imaginary elliptic cylinder	2	3		yes
Hyperbolic cylinder	2	3		no
Real intersecting planes	2	2		no
Imaginary intersecting planes	2	2		yes
Parabolic cylinder	1	3		
Imaginary parallel planes	1	2		
Coincident planes	1	2		
Real parallel planes	1	1		

To find the position and orientation of the quadric surface, let us consider a real ellipsoid centered about the origin (body coordinates).

$$\frac{\tilde{x}^2}{r_1^2} + \frac{\tilde{y}^2}{r_2^2} + \frac{\tilde{z}^2}{r_3^2} - 1 = 0 \qquad (20)$$

or

$$\begin{bmatrix} \tilde{x} & \tilde{y} & \tilde{z} & 1 \end{bmatrix} \begin{bmatrix} \frac{-1}{r_1^2} & 0 & 0 & 0 \\ 0 & \frac{-1}{r_2^2} & 0 & 0 \\ 0 & 0 & \frac{-1}{r_3^2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ 1 \end{bmatrix} = 0 \qquad (21)$$

Translating and rotating the cylinder, the position of the ellipsoid with respect to world coordinates is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} + \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \qquad (22)$$

where $\mathbf{R}$ is a 3x3 rotation matrix and (x_o, y_o, z_o) is the translated position of the ellipsoid. Solving this equation for $\tilde{x}, \tilde{y}, \tilde{z}$ and substituting into Equation (21), we can show that the equation for the ellipsoid with respect to the world coordinates is

$$f(x,y,z) = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} & & & 0 \\ & \mathbf{D}' & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$-2\begin{bmatrix} x_o & y_o & z_o \end{bmatrix} \mathbf{D}' \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} x_o & y_o & z_o \end{bmatrix} \mathbf{D}' \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + 1$$

$$(23)$$

where $\mathbf{D}' = \mathbf{R} \begin{bmatrix} \frac{-1}{r_1^2} & 0 & 0 \\ 0 & \frac{-1}{r_2^2} & 0 \\ 0 & 0 & \frac{-1}{r_3^2} \end{bmatrix} \mathbf{R}^T$.

Compared to the quadric equation, notice that the eigenvalues of $\mathbf{D}'$ are the inversely proportional to the ellipsoid radii squared, and that the

5

eigenvectors of $\mathbf{D}'$ compose the rotation matrix $\mathbf{R}$.

Normalizing the constant terms in Equation (23) by letting

$$L = \begin{bmatrix} x_o & y_o & z_o \end{bmatrix} \mathbf{D}' \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + 1 \qquad (24)$$

we find that $\mathbf{D} = \dfrac{\mathbf{D}'}{L}$.

From the second term on the right hand side of Equation (23), the position of the ellipsoid in world coordinates is related to the quadric parameters u, v, and w by

$$\begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} = -\mathbf{D}^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad (25)$$

Care must be taken when $\mathbf{D}$ is singular. This will occur when there is a single translation along only one axis. These special cases can be identified by examining the elements of $\mathbf{D}$.

From Equations (24) and (25), it can be shown that

$$L = \left(1 + \begin{bmatrix} x_o & y_o & z_o \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}\right)^{-1} . \qquad (26)$$

Finally, the radii of the ellipsoid are given by

$$r_j = \sqrt{\frac{-1}{L\lambda_j}} \quad , j = 1,2,3 \qquad (27)$$

Therefore, the orientation of the ellipsoid is determined from the eigenvectors of $\mathbf{D}$, the position is determined from Equation (25), and the radii are determined from Equations (26) and (27).

The analysis for a real elliptic cylinder is the same except that radius r_3 is zero. The height of the cylinder is determined by projecting all data points into the body coordinate frame and

looking at the maximum and minimum points along the z-axis. This same type of analysis can be applied to several of the remaining 15 quadric surfaces, including real quadric cones and hyperboloids.

4. Experimental Results

The above algorithms were tested on real data sets from a laser range finder and a structured lighting system. The next two figures area taken from range data supplied by Coleman Research Corporation from their laser range finder. This test case shows a corner with a cylinder and a box. Figure 2 has approximately 48,000 range data points. A simple triangulation of this data would generate around 96,000 polygons. Figure 3 shows the results of planar fitting (the box and walls) and quadric fitting (the cylinder). The polygon count for this image is around 150. The additional polygons are kept to preserve perimeter details on the planar surfaces. The cylinder is represented with 24 faces. If perimeter was not preserved, this image could easily be reduced to 24 triangles, which is a 4000 to 1 data reduction.
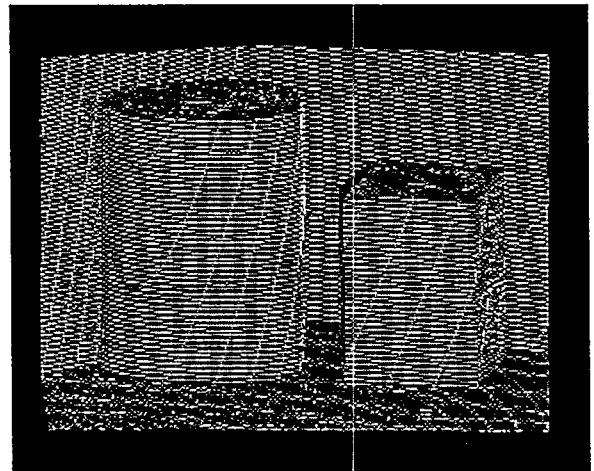

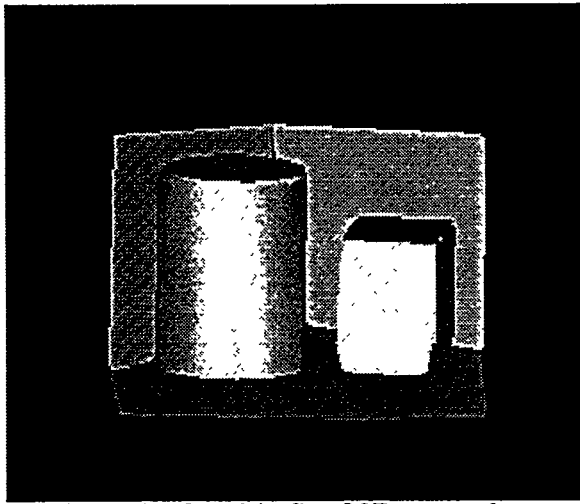Figure 2. Range data displayed as 3D points.

6

Figure 3. Quadric fit of 3D range data.

In addition to data reduction, primitive fitting can be fairly insensitive to noise; particularly the statistical noise seen along broad surfaces. Bad readings, typically caused by reflectance problems, are usually easily removed during the segmentation step.

Figure 4 is an example of a real world scene taken from Sandia's structured lighting system. The intensity image shows a 6 foot diameter bowl, filled with vermiculite. It also contains a large cylindrical piece, and on the outer edges are two flat plates: one wood and one steel. The model image shows the results of surfacing from range data. The uneven surface of the vermiculite does not lend itself to primitive fitting without a severe reduction in surface location accuracy. However, the cylinder and plates can be modeled with high data reduction and minimal loss of surface accuracy.
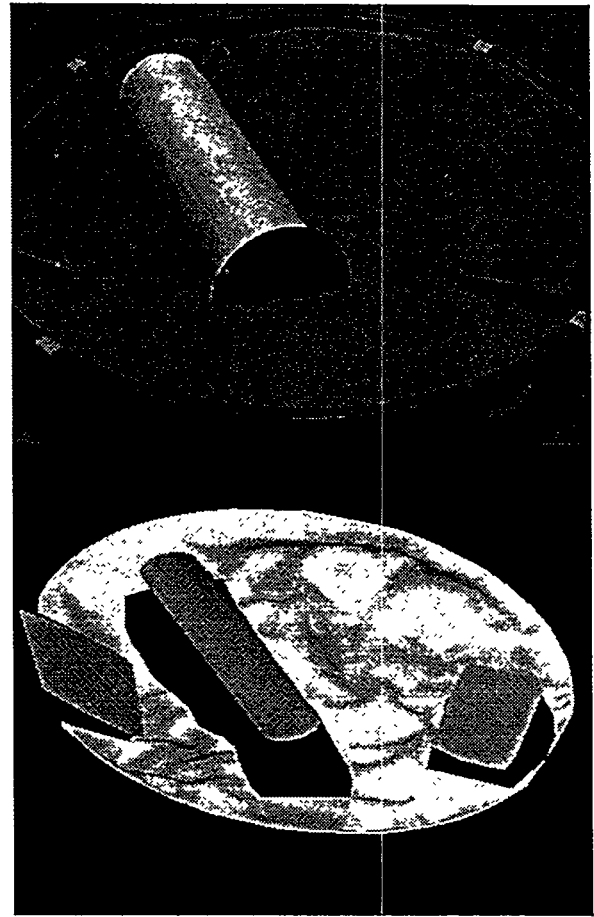

Figure 4. Intensity image and modeled image.

5. Conclusion

Two least squares algorithms were presented for fitting both planar and quadric surfaces to three dimensional scatter point data. From the eigenvalues of the quadric parameters, it is possible to distinguish between cylinders, ellipsoids, cones, and hyperboloids. The eigenvalues and eigenvectors can then be used to determine the position, orientation, height, and diameter of these surfaces. These algorithms have been successfully tested on data from both a laser range finder and a structured lighting system. It should be noted that these images were segmented by hand and then fit. Tests showed that outliers in the data can result in selecting the wrong type of quadric. In these cases, the operator had to go back and remove the outliers. Future work is needed to automatically segment the images and to remove outliers before least squares fitting of the data.

7

References

[1] Barry, R.E., Little, C.Q., and Burks, B.L., "Requirements and Design Concept for a Facility Mapping System," *Proc. ANS 6th Topical Meeting on Robotics and Remote Systems*, Monterey CA, pp. 775-783, February 5-10, 1995.

[2] Little, C.Q., and Wilson, C.W., "Rapid World Modelling for Robotics," *Proc. World Automation Congress '96*, May 1996.

[3] Faux, I.D. and Pratt, M.J., *Computational Geometry for Design and Manufacture*, Ellis Horwood Limited, England, 1985.

[4] Limaiem, A., and ElMaraphy, H.A., "Automatic Inference of Parametric Equations in Geometric Modeling Using Dual Kriging," *Proc. 1996 IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, pp. 1499-1504, 1996.

[5] Garcia, M.A., and Basanez, L., "Efficient Free-Form Surface Modeling with Uncertainty," *Proc. 1996 IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, pp. 1825-1830, 1996.

[6] Garcia, M.A., "Fast Approximation of Range Images by Triangular Meshes Generated through Adaptive Randomized Sampling," *Proc. 1995 IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, 1995.

[7] Johnson, A., Leger, P., Hoffman, R. Hebert, M., and Osborn, J., "3-D Object Modeling and Recognition for Telerobotic Manipulation," *Proc. IEEE Intelligent Robots and Systems*, pp. 103-110, August 1995.

[8] Beyer, W.H., *Standard Mathematical Tables*, 24th Edition, CRC Press, Inc., West Palm Beach, Florida, p. 310, 1978.