Uncertainty in Object Pose Determination with Three Light-Stripe Range Measurements

Keiichi Kemmotsu[1] Takeo Kanade

January 1993

CMU-CS-93-100

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was sponsored by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. government.

[1] This research was performed while the first author was with Carnegie Mellon University. His current address is Advanced Technology Research Center, Mitsubishi Heavy Industries, Ltd., 1-8-1 Sachiura Kanazawa-ku, Yokohama, 236, Japan.

Keywords: computer vision, object recognition, interpretation tree, geometric constraints, pose determination, geometric uncertainties

Abstract

The pose (position and orientation) of a polyhedral object can be determined with sparse range data obtained from simple light-stripe range finders. However, the sensing data inherently contains some error which introduces uncertainty in the determination of the object's pose. This paper presents a method for estimating the uncertainty in determining the pose of an object when using several light-stripe range finders. Three dimensional line segments obtained by the range finders are matched to model faces based on an interpretation tree search. The object pose is obtained by a least squares fit of the segment-face pairings. We show that the uncertainty in the position of the object can be estimated using the covariance matrix of the endpoint positions of the sensed line segments. Experiments with three light-stripe range finders show that our method makes it possible to estimate how accurately the pose of an object can be determined.

Contents

1 Introduction

Recognizing the pose of a three-dimensional (3-D) object in a workspace is a fundamental task in many computer vision applications, including automated assembly, inspection, and bin picking. Many object recognition algorithms have been developed. However, there has been little attention given to estimating the uncertainty of object pose determinations. In this paper, we study a problem of estimating uncertainty in determining the pose of a polyhedral object when using multiple light-stripe range finders.

Simple light-stripe range finders are among the fastest and least expensive ways to acquire accurate range data. Multiple range finders viewing an object from different perspectives can usually provide enough constraints to determine the object's pose. Imagine that a polyhedral object is placed at an arbitrary pose in the workspace and that we place three simple light-stripe range finders above the workspace. Based on an interpretation tree search technique, 3-D line segments obtained by the range finders can be assigned to model faces consistent with geometric constraints. Once a feasible interpretation is found that satisfies the geometric constraints for all line segments, the transformation from the model coordinate frame to the world coordinate frame is obtained by a least squares method.

As a result of sensing error, the transformation contains inaccuracies. Therefore, we need to estimate uncertainty in determining the pose of an object. Using an error analysis based on the convergence properties of the least squares fit, we obtain a relationship between the covariance matrix of the line segments' endpoint positions and the covariance matrix of the position of each object vertex. The pose uncertainty of the object can then be estimated from this relationship.

Related Work

Our object recognition method is based on the use of simple light-stripe range finders. Though many 3-D object recognition systems using range image information have been reported [2], [5], [6], [7], [16] and some range imaging techniques are very fast [1], the recognition processes of these systems are still very slow, making such techniques impractical for industrial applications. Recognition is slow because these systems extract many surfaces and/or edges from raw, dense range images; this process is time-consuming and sometimes generates incorrect features, which cause difficulty when matching the features to object models. While a dense range image is appropriate to describe a complex scene precisely, scenes in industrial applications can usually be simplified by modifying the environment to enable object recognition using only simple sensors such as light-stripe range finders.

It has already been shown that light-stripe range finders are effective in determining the pose of polyhedral objects in controlled environments where some information about the object's pose is already known. Gordon and Seering [8] showed that object pose can be determined precisely with one simple light-stripe range finder providing that the *a priori* pose of the object is known approximately. Chen [3] proposed a pose determination method with three known correspondences between line segments and model faces.

Before we determine the pose of an object, we must first determine feature correspondences. To find correspondences between sensed features and model features, an interpretation tree search method with geometric constraints is used. Grimson and Lozano-Pérez [9] demonstrated that local unary and binary geometric constraints are very effective in reducing the size of an interpretation tree. While Grimson and Lozano-Pérez used the position and iocal surface orientation of a small set of points on the object, Murray and Cook [14] presented geometric constraints for sensed edges corresponding to model edges. However, since a light-stripe range finder provides the position and direction of a 3-D line segment that lies on an object face, different geometric constraints are required.

A least squares method is usually used to determine the pose of an object, that is, to obtain the rotation and translation components of a transformation [6], [12]. Grimson [10] suggested that uncertainty bounds on the object pose can be tightened by propagating initial errors algebraically through interpretation equations. Ellis [4] showed that the uncertainty bounds can be tightened by considering the cross-coupling between rotational and translational uncertainties. In [4] and [10], sensed surface normals were used to estimate the upper limit of the transformation error. A weighted least squares method for determining an error bound on the orientation of an object by considering the contribution to the error from each sensed vertex was shown in [18]. A criterion for choosing measurement points to minimize transformation error by using a sensitivity matrix was discussed in [17]. Since the pose uncertainty of an object can be represented by the covariance matrix of the position of each object vertex, we explore a pose uncertainty estimation method that uses the covariance matrix of the endpoint positions of sensed line segments.

In this section, we introduced the research objective, and reviewed related work. In Section 2 an interpretation tree search technique with geometric constraints suitable for line segments is discussed. In Section 3 we focus on the error analysis for object pose determination and describe a pose uncertainty estimation technique. In Section 4, experiments with three light-stripe range finders show that our object recognition method successfully determines the pose of an object and that our pose uncertainty estimation method provides a useful tool for estimating how accurately the position and orientation of an object can be determined.

2 Fast Object Recognition with Three Light-Stripe Range Measurements

The task of model-based object recognition is to match sensed features to model features and to determine the object pose in a 3-D world coordinate frame. We begin with an example

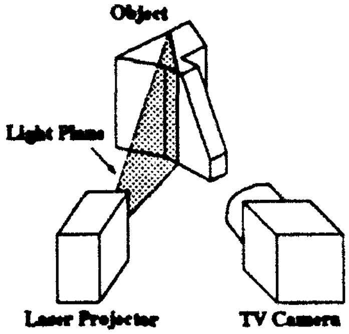


Figure 1: A simple light-stripe range finder.

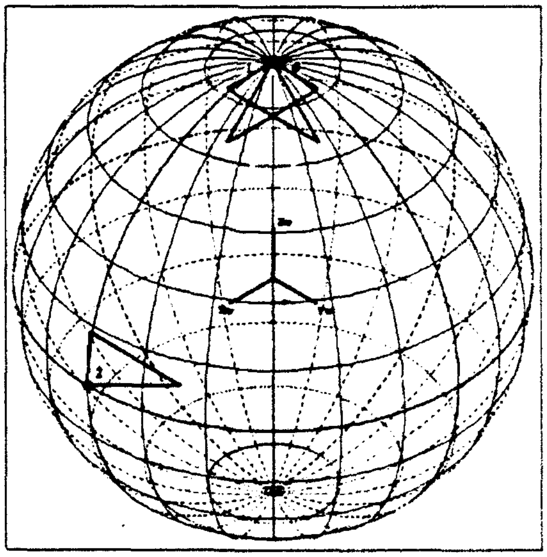


Figure 2: Sensor placement for object recognition. Sensors 0 and 1 are placed on the z axis, directed toward the origin. Their sensing planes, which are displayed as triangles, perpendicularly intersect. Sensor 2 is placed on the x axis and its sensing plane lies on the x-y plane.

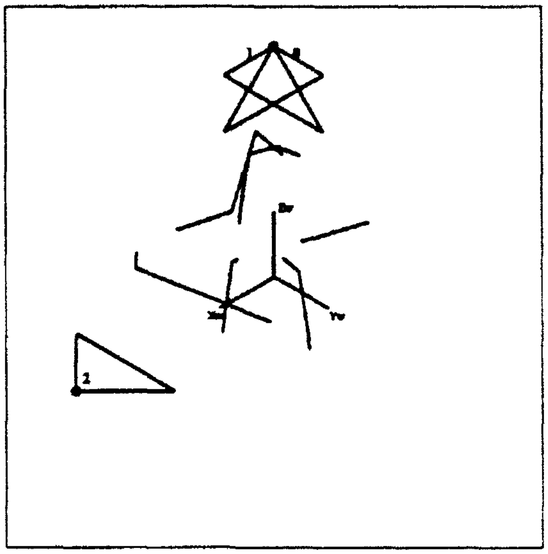
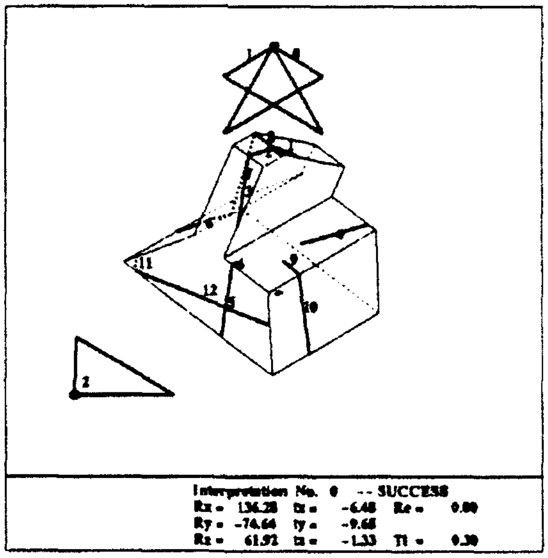


Figure 3: Obtained 3-D line segments on object faces.



```
Interpretation No. 0   -- SUCCESS
Rx =  136.28  tx =   -6.48  Re =    0.00
Ry =  -74.64  ty =   -9.68
Rz =   61.92  tz =   -1.33  Ti =    0.30
```

Figure 4: An object recognition result. Estimated transformations $\omega(R_x)$, $\varphi(R_y)$ and $\kappa(R_z)$ are given in degrees and t_x, t_y and t_z are given in millimeters. R_e is the standard deviation of the distances between the endpoints of the line segments and the corresponding object faces. T_i shows the elapsed time in seconds (Sun SPARCstation IPC).

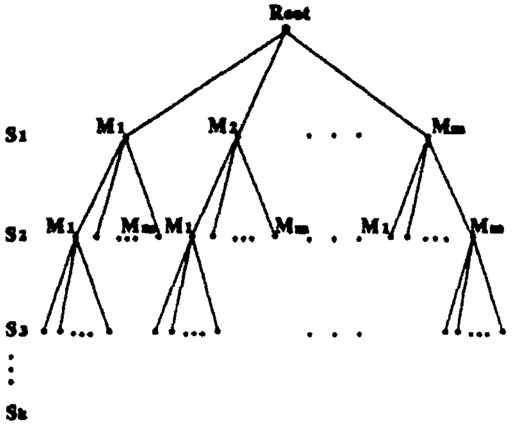


Figure 5: An interpretation tree for assignments between k sensed features and m model features.

of recognizing an object. A simple light-stripe range finder projects a light plane onto the faces of an object and measures the 3-D line segments created by the light-stripe as shown in Figure 1. Three identical range finders are placed in the world coordinate frame as shown in Figure 2. We assume that the light source and viewpoint of each range finder are coincident. The range finders obtain 3-D line segments as shown in Figure 3. Our matching scheme by an interpretation tree search assigns the sensed line segments to the corresponding model faces and uses geometric constraints to eliminate inconsistent segment-face pairings. The object's pose is successfully determined as shown in Figure 4. In this section, we describe our object recognition and pose determination technique.

2.1 Interpretation Tree Search by Geometric Constraints

Let S_1, S_2, ... , S_k denote sensed line segments and M_1, M_2, ... , M_m denote model faces. In general, there are m^k ways of matching the line segments to the model faces assuming that each line segment must match to one model face. Though such assignments can be represented by an interpretation tree as shown in Figure 5, it is not feasible to explore the entire tree to find consistent interpretations. Rather, geometric constraints are used to discard inconsistent pairings while searching the tree in a depth-first and backtracking manner.

Grimson and Lozano-Pérez [9] showed that the interpretation tree search technique with local unary and binary geometric constraints is a useful method to find a consistent set of pairings (S_1, M_{p_1}), (S_2, M_{p_2}), ..., (S_k, M_{p_k}) where M_{p_i} is the model face which corresponds to line segment S_i. The unary constraints check the consistency of a pairing between a line segment and a model face and the binary constraints check the consistency of two pairings. The specific constraints used in our method are given in Appendix A.

These unary and binary constraints are weaker than those in Grimson's work[11] which are based on face matching, since line segments carry less information than faces. Therefore, after applying the unary and binary constraints, we apply triplet constraints which check a triplet of pairings between line segments and model faces to prune the interpretation tree more efficiently. As deeper nodes are reached in the interpretation tree, more possible triplet pairings exist making a triplet constraint check appear to be time-consuming[15]. To speed the process, we choose three line segments and three model faces under the condition that two of the line segments must intersect each other. Since the two line segments are therefore coplanar, two of the three model faces must be the same. The intersecting line segments can define the normal of the model face on which the line segments lie. The normal of the other model face can be obtained by solving a quadratic equation since the normal must be perpendicular to the direction vector of the third line segment. Further details of the triplet constraints may be found in Appendix B.

2.2 Ordering Sensed Features

The order in which sensed features are matched is very important since early rejection of inconsistent nodes results in more efficient pruning of the interpretation tree. In our recognition algorithm, intersecting line segments play an important role in the tree search because the triplet constraints can be applied to such line segments to rapidly eliminate segment-face pairings that cannot be consistent. Intersecting line segments should therefore be used as early as possible to rapidly prune the interpretation tree and save computation time.

2.3 Computing Transformations

2.3.1 Rotation Component

Intersecting line segments are not only used in the triplet constraints, but also to compute the rotation matrix $\boldsymbol{R}$ of the transformation from the model coordinate frame to the world coordinate frame (see Appendix B).

If there are no intersecting line segments, a numerical polynomial-based technique is used to calculate the transformation after at least three consistent pairings between line segments and model faces are found. Let $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{w}$ denote the unit direction vectors of the three line segments and let $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ denote the unit normal vectors of the corresponding model faces. We can calculate a rotaticn matrix, such that the rotated vectors $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ of the vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are orthogonal to $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{w}$ respectively. Chen [3] has presented a similar polynomial approach to solve the same problem through a canonical configuration to reduce the number of unknowns to two. It is important to note that there are certain conditions that must be satisfied by the configuration of the surface normals and the direction vectors to solve this problem.

Unfortunately, these general polynomial-based methods are very sensitive to noise as well as computationally expensive ince an eighth-degree equation must be solved. On the other hand, our method which uses intersecting line segments is very fast and robust since a transformation is obtained by solving a quadratic equation in the triplet constraint check. Polynomial-based methods are therefore used only in the rare cases in which no intersecting line segments exist.

2.3.2 Translation Component

Next, we solve the translation component $\boldsymbol{t}$ of the transformation. A point $\boldsymbol{p}$ in the world coordinate frame is related to a corresponding point $\boldsymbol{P}$ in the model coordinate frame

$$\boldsymbol{p} = \boldsymbol{R}\boldsymbol{P} + \boldsymbol{t}. \tag{1}$$

Suppose that a line segment S_i, whose endpoints are $\boldsymbol{b}_i$ and $\boldsymbol{e}_i$, corresponds to a model face M_{p_i}. Any point $\boldsymbol{X} = (X, Y, Z)^T$ on the model face satisfies the equation

$$\boldsymbol{N}_{p_i}^T \boldsymbol{X} + D_{p_i} = 0 \tag{2}$$

where $\boldsymbol{N}_{p_i}$ and D_{p_i} are the unit normal and offset of the model face M_{p_i} respectively. If the point $\boldsymbol{p}$ is on the line segment S_i, the squared distance from the point to the corresponding model face is given by

$$(\Delta d_i)^2 = \left(\boldsymbol{N}_{p_i}^T (\boldsymbol{R}^{-1}(\boldsymbol{p} - \boldsymbol{t})) + D_{p_i}\right)^2. \tag{3}$$

The translation component $\boldsymbol{t}$ is therefore obtained by minimizing the sum of the integral of the squared distance along each line segment over all pairings of an obtained feasible interpretation (S_i, M_{p_i}) for $i = 1, \ldots, k$

$$E = \sum_{i=1}^{k} \int_{\boldsymbol{b}_i}^{\boldsymbol{e}_i} (\Delta d_i)^2 ds_i \tag{4}$$

where ds_i is an element of line segment S_i.

If the residual E of fitting the model faces to the line segments is small enough, and if the endpoints of line segments S_i for $i = 1, \ldots, k$ pass the additional test that they are near to the model face M_{p_i}, then this interpretation is regarded as a globally consistent interpretation.

2.3.3 Refining the Transformation

After an interpretation is deemed globally consistent, the rotation and translation components of the transformation are improved by another least squares process. Both initial rotation and translation values are used simultaneously to refine the fit of the sensed line segments to the model faces.

Table 1: Recognition results for 1000 trials.

Conditions	Successful trials	Failed trials	Recognition time (sec)
Unary & binary constraints No triplet constraints No feature ordering	895	105	10.1
Unary & binary constraints Triplet constraints No feature ordering	949	51	0.7
Unary & binary constraints Triplet constraints Feature ordering	949	51	0.1

2.4 Simulation

We run simulation to test the effectiveness of our object recognition method. We use a polyhedral object as shown in Figure 1. Three hypothetical light-stripe range finders are placed in the world coordinate frame as shown in Figure 2. The object is then randomly located in the world coordinate frame. A simulation proceeds as follows:

- As input data for the recognition program, a range finder simulator calculates the line segments which the three light-stripe range finders would get from viewing the object.

- Feasible interpretations are obtained by performing the interpretation tree search with the geometric constraints.

- Each feasible interpretation is verified by comparing object vertices found using the recognition algorithm with the correct values. If all estimated positions of the vertices are near enough to corresponding correct positions, the interpretation is regarded as correct.

The results of 1000 trials are shown in Table 1. All failed trials correspond to multiple interpretations which include some correct and some incorrect interpretations. Adding the triplet constraints reduces the average recognition time to 0.7 seconds and the number of failed trials to half. The triplet constraints are very efficient not only in pruning the interpretation tree, but in improving recognition performance.

The ordering of line segments is also important. A typical example is shown in Figures 4 and 6. The intersecting line segments No.5 and No.12 in Figure 4 play a crucial role to decrease the number of nodes of the interpretation tree. As a result of ordering the line

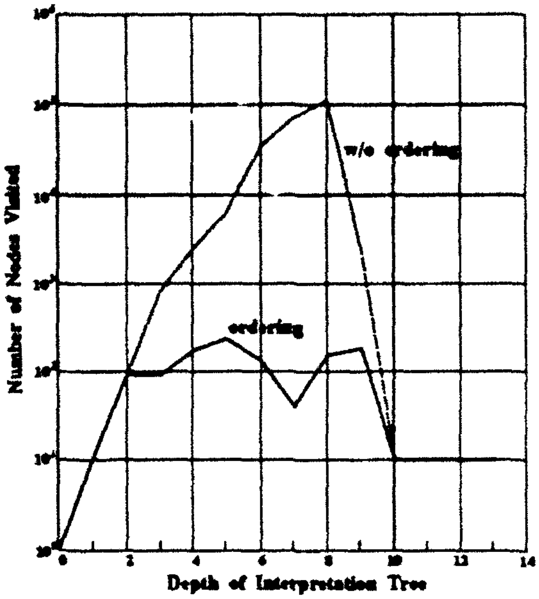


Figure 6: The number of nodes visited in the interpretation tree without feature ordering and with feature ordering. Ordering line segments for the tree search dramatically speeds pruning the interpretation tree.

segments so that the intersecting line segments are examined first, the computation time is decreased from 20 seconds to 0.3 seconds.

One problem with this recognition technique is that it takes a long time to recognize an object if there are no intersecting line segments. In most trials, however, intersecting line segments appear on object faces, which is a characteristic when using multiple range finders. As a result, the average computation time for object recognition is about 0.1 second.

3 Geometric Uncertainties in Pose Determination

Now we can determine the pose of an object. However, due to sensing error inherent in measuring line segments, the obtained transformation contains some error, which causes uncertainty in the position of the object. This section describes our technique for estimating the pose uncertainty.

3.1 Uncertainty

The object pose itself is obtained by minimizing the sum of the squared distances between sensed line segments and corresponding object faces, and hence the transformation error is defined as a perturbation around the correct transformation with respect to the sensing error.

Let the rotation component $\boldsymbol{R}$ and translation component $\boldsymbol{t}$ of the transformation be denoted by

$$\boldsymbol{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{pmatrix} \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix} \begin{pmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{5}$$

where ω, φ and κ are rotation angles around x, y and z axes in the world coordinate frame. If $\boldsymbol{x} = (t_x, t_y, t_z, \omega, \varphi, \kappa)^T$ denotes the six transformation variables, the transformation error is defined by

$$\Delta\boldsymbol{x} = (\Delta t_x, \Delta t_y, \Delta t_z, \Delta\omega, \Delta\varphi, \Delta\kappa)^T.$$

In addition, we define the sensing error of the endpoints $(x_{2i-1}, y_{2i-1}, z_{2i-1})$ and (x_{2i}, y_{2i}, z_{2i}) of line segment S_i as

$$\Delta\boldsymbol{s} = (\Delta x_1, \Delta y_1, \Delta z_1, \ldots, \Delta x_{2k}, \Delta y_{2k}, \Delta z_{2k})^T.$$

3.2 Relationship between Sensing Error and Transformation Error

Our object recognition technique finds pairings (S_i, M_{p_i}) between line segments and model faces. Due to sensing error, a point $\boldsymbol{p}$ on a line segment S_i lies off the corresponding object face M_{p_i} by a distance Δd_i given by equation (3). As we mentioned in Section 2, the pose of the object is determined minimizing the residual E of equation (4) in terms of $\boldsymbol{x}$. The necessary condition for E to reach an extremum is given as

$$\frac{\partial E}{\partial t_x} = \frac{\partial E}{\partial t_y} = \frac{\partial E}{\partial t_z} = \frac{\partial E}{\partial \omega} = \frac{\partial E}{\partial \varphi} = \frac{\partial E}{\partial \kappa} = 0. \tag{6}$$

Now to examine the uncertainty in the transformation caused by sensing error, we linearize these non-linear equations around the approximate solution $(\boldsymbol{x}_0, \boldsymbol{s}_0)$ which corresponds to the correct transformation and endpoints,

$$\boldsymbol{A}\Delta\boldsymbol{x} \cong -\boldsymbol{B}\Delta\boldsymbol{s} \tag{7}$$

where $\boldsymbol{A}$ is the Hessian matrix of E with respect to $\boldsymbol{x}$ and $\boldsymbol{B}$ is the Jacobian matrix of $\frac{\partial E}{\partial \boldsymbol{x}}$ with respect to $\boldsymbol{s}$.

Then we relate the object vertex position error to the transformation error $\Delta\boldsymbol{x}$. The position of a vertex $\boldsymbol{v}_j$ in the world coordinate frame is related to a vertex $\boldsymbol{V}_j$ in the model coordinate frame by

$$\boldsymbol{v}_j = \boldsymbol{R}\boldsymbol{V}_j + \boldsymbol{t}. \tag{8}$$

The position error $\Delta\ \boldsymbol{v}_j$ is then given by

$$\Delta\boldsymbol{v}_j \cong \boldsymbol{D}_j\Delta\boldsymbol{x} \tag{9}$$

where $\boldsymbol{D}_j$ is the Jacobian matrix of $\boldsymbol{v}_j$ with respect to $\boldsymbol{x}$. By substituting equation (7) into equation (9), the relationship between the position error and the sensing error becomes

$$\Delta \boldsymbol{v}_j \cong -\boldsymbol{D}_j \boldsymbol{A}^{-1} \boldsymbol{B} \Delta \boldsymbol{s}. \tag{10}$$

The covariance matrix $\boldsymbol{C}_{v_j}$ of the vertex $\boldsymbol{v}_j$ is given by

$$\begin{aligned} \boldsymbol{C}_{v_j} &\equiv \boldsymbol{E}(\Delta \boldsymbol{v}_j \Delta \boldsymbol{v}_j^T) \\ &= \boldsymbol{D}_j (\boldsymbol{A}^{-1}\boldsymbol{B}) \boldsymbol{C}_s (\boldsymbol{A}^{-1}\boldsymbol{B})^T \boldsymbol{D}_j^T \end{aligned} \tag{11}$$

where $\boldsymbol{C}_s$ is the covariance matrix of the line segments' endpoint positions. The elements of the covariance matrix $\boldsymbol{C}_{v_j}$ show how uncertain the vertex position is, and hence the x, y and z components of the position error of each vertex can be approximated as

$$(\Delta v_{j_x}, \Delta v_{j_y}, \Delta v_{j_z}) = (\sqrt{C_{v_{j11}}}, \sqrt{C_{v_{j22}}}, \sqrt{C_{v_{j33}}}). \tag{12}$$

3.3 Examples

The following are some examples of estimating the uncertainty in pose determination. Given the shape of an object, a transformation $\boldsymbol{x}$ for the object and a placement of three light-stripe range finders, a range finder simulator calculates line segments which would appear on the object. We assume that all endpoints of obtained line segments have the same error (zero mean Gaussian white noise $N(0,1)$) and that any two endpoints are independently measured and their respective errors are not related (though the mechanism of the sensing error of a range finder is complex in practice [13]). Thus, the covariance matrix $\boldsymbol{C}_s$ of the endpoint positions of the line segments becomes the identity matrix. We can estimate the uncertainty of each vertex of the object with equation (11).

Given a model as shown in Figure 1, a sensor placement as in Figure 2, and the same transformation as in Figure 4, an estimated uncertainty on each vertex of the object is shown in Figure 7. In this figure, the lengths of three bars on each vertex along x, y, z directions are given by equation (12), and show how uncertain the position of each vertex is.[1] The position error depends on the pose of the object with respect to the range finders, that is, the spatial distribution of line segments on the object faces. Another example with a different transformation is shown in Figure 8. Figure 9 shows the object in the same pose as Figure 8, but with a different sensor placement. The position error in Figure 9 is much larger than that in Figure 8 as a result of the line segment distribution on the object faces.

In general, as the number of different faces on which line segments fall increases, pose determination accuracy also increases. Note that position error cannot be estimated by

[1] For display purpose, those lengths equal $12\Delta v_{j_x}$, $12\Delta v_{j_y}$, and $12\Delta v_{j_z}$ respectively.

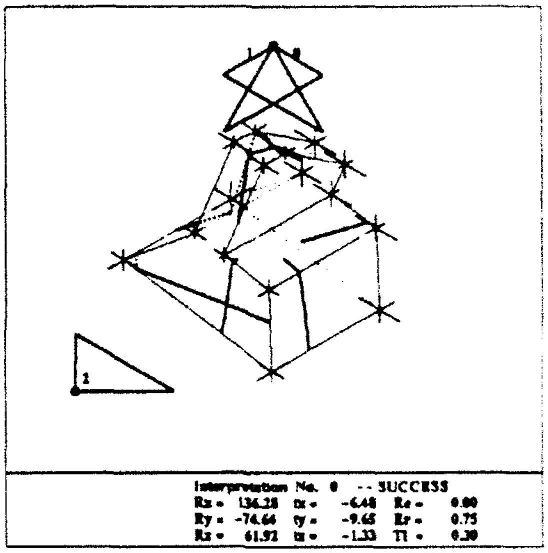

1
Interpretation No. 0 -- SUCCESS
Rx = 136.28 tx = -6.48 Re = 0.00
Ry = -74.64 ty = -9.65 Er = 0.75
Rz = 61.92 tz = -1.33 Tl = 0.30

Figure 7: An uncertainty estimation result after recognizing the object. Three bars on each vertex show the uncertainty in pose determination. E_r(mm) is the average position error of all vertices.

the method described here when the surface normals of all the object faces on which line segments lie are coplanar. In this case, the translation component cannot be estimated by the pose determination technique because there is an unconstrained degree of freedom.

4 Experiments

This section presents experimental results in recognizing an object and estimating pose uncertainty. First, the procedure of our experiments, which include image processing for extracting 2-D line segments and computing the positions of 3-D line segments in the world coordinate frame, is described. Then, experimental results are shown and compared with simulation results.

4.1 Procedure for Experiments

Each light-stripe range finder is composed of a TV camera with a 16mm lens and a laser diode projector whose wavelength is 670 nm. The laser beam is spread by a cylindrical lens to generate a light plane. The baseline length between the TV camera and the laser projector is about 100 mm. We place three identical range finders above the workspace as shown in Figure 10. The distance between each range finder and the workspace center is about 350 mm and each range finder's absolute accuracy of measuring 3-D coordinates is ± 0.5 mm within the workspace.

For each range finder, line segments are extracted by the following procedure:

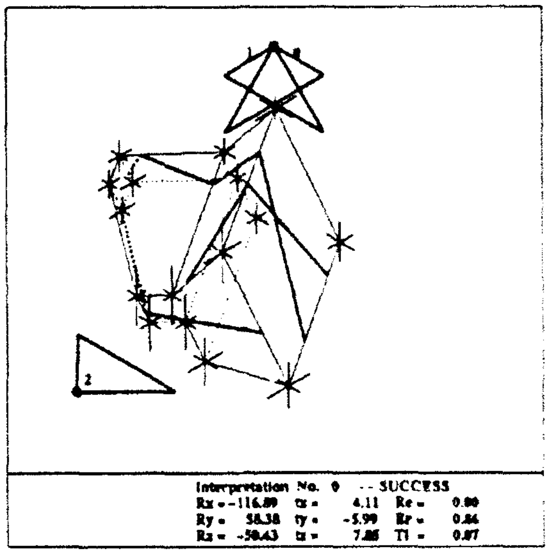


Figure 8: An uncertainty estimation result with the same sensor placement as in Figure 7 but a different object pose.

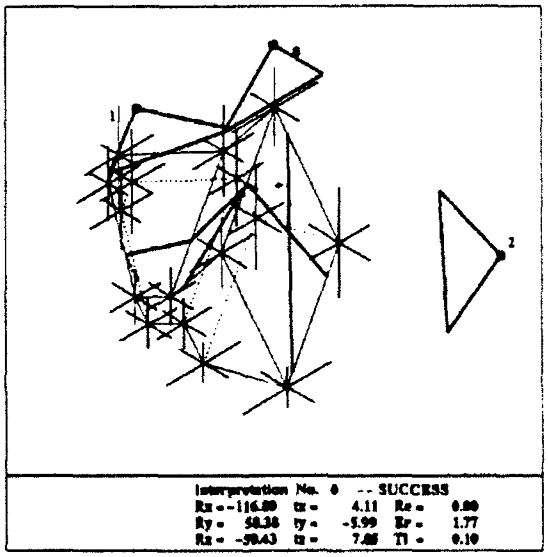


Figure 9: An uncertainty estimation result with the same object pose as in Figure 8 but a different sensor placement.

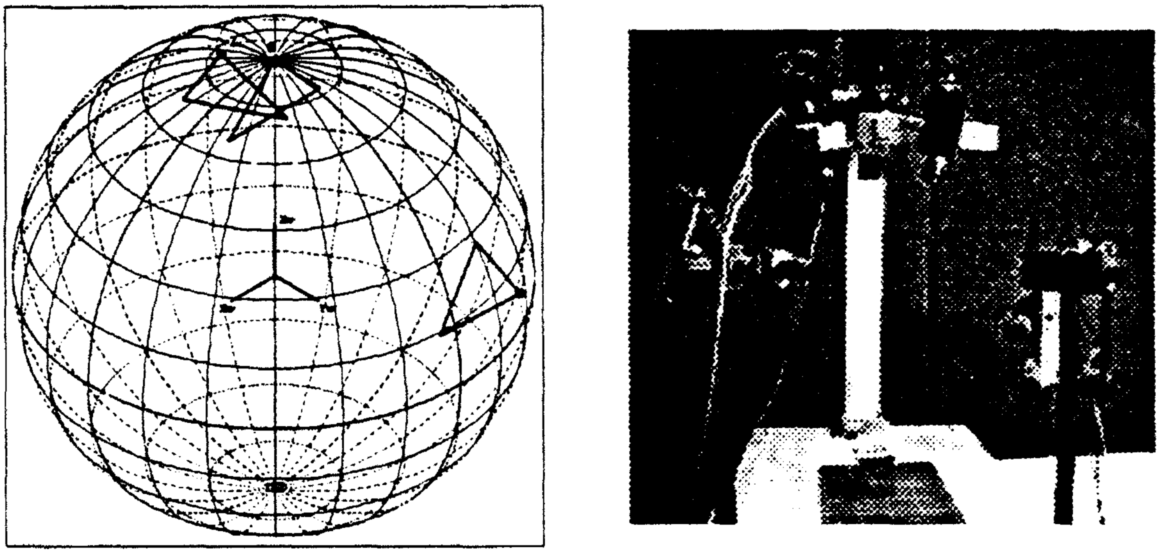


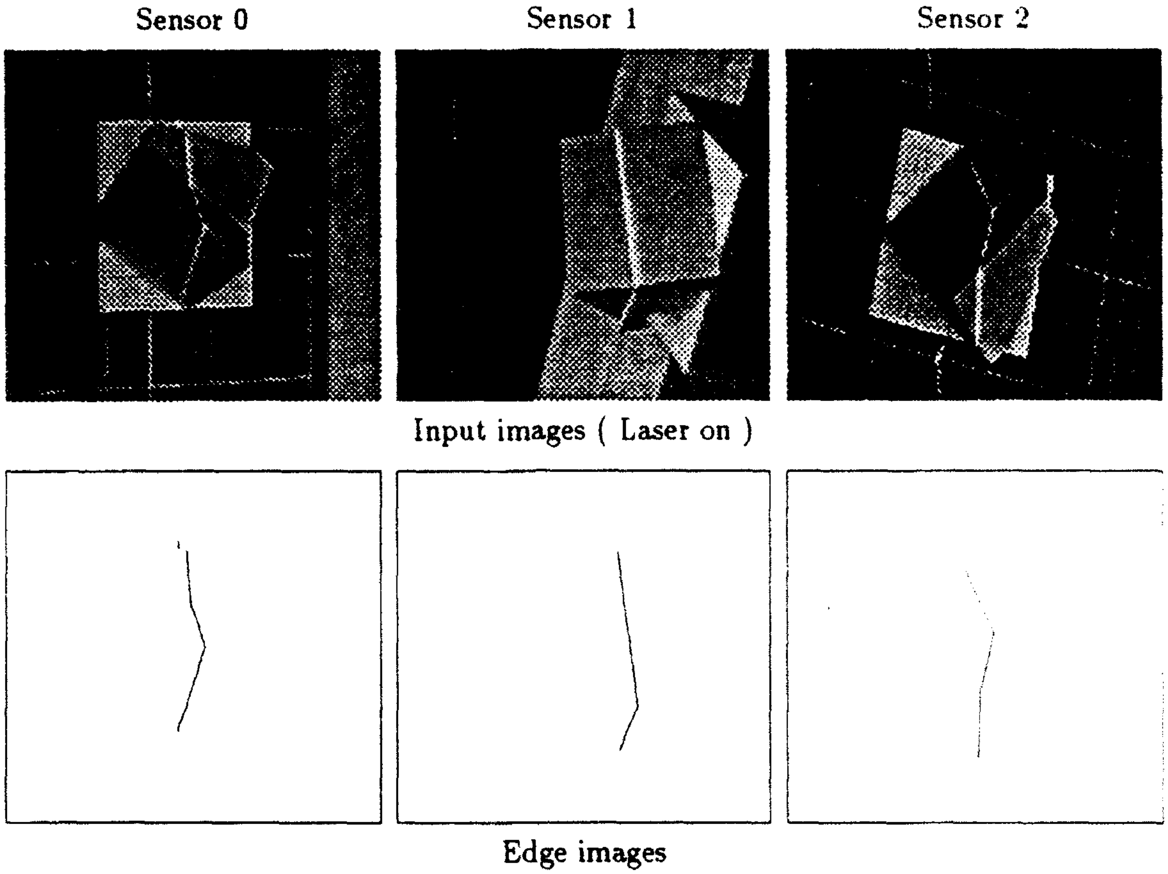
Figure 10: Sensor placement for experiments.

- Take an image with the laser diode projector off.
- Take an image with the laser diode projector on.
- Compute image differences and detect edges.
- Track those edges and find the endpoints of the 2-D line segments.
- Compute the positions of 3-D line segments using projective transformation coefficients (coefficients are calculated during calibration).

Once all the 3-D line segments have been found, apply the object recognition technique. Finally, estimate the uncertainty of each calculated vertex position.

4.2 Experimental Results

An object like the one depicted in Figure 1 is placed at an arbitrary pose in the workspace. Each range finder takes two images (one with the laser diode on, one with the diode off) and detects edges as shown in Figure 11. Figure 12 shows obtained 3-D line segments and object recognition and position error estimation results. For comparison, Figure 13 shows a simulation result with the same object pose under the same sensor placement as the experiment shown in Figure 12. The recognition time in the experiment is 0.67 sec, while only 0.05 sec in the simulation. In the experiment, the geometric constraints used in the interpretation tree search were weakened to allow for error in the measurement, thus, increasing the number of


Sensor 0 Sensor 1 Sensor 2

Input images (Laser on)

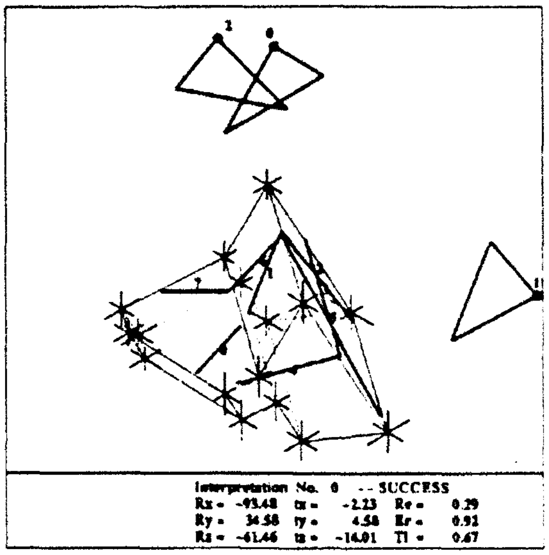
Edge images

Figure 11: Input images for the three light-stripe range finders and extracted edge images. The object is placed on a support cube whose size is 60 × 60 × 60 mm. The cube is not regarded as a part of the object.

visited nodes. Note that the line segments No. 0 and No. 1 and the line segments No. 6 and No. 7 in Figure 12 are not connected. Edge tracking often fails to detect a correct junction of two line segments on a concave object edge as a result of interreflection of the light plane. Nevertheless, recognition succeeded because our matching technique uses assignments of line segments to model faces instead of relying on exact matching of line segment endpoints to model edges.

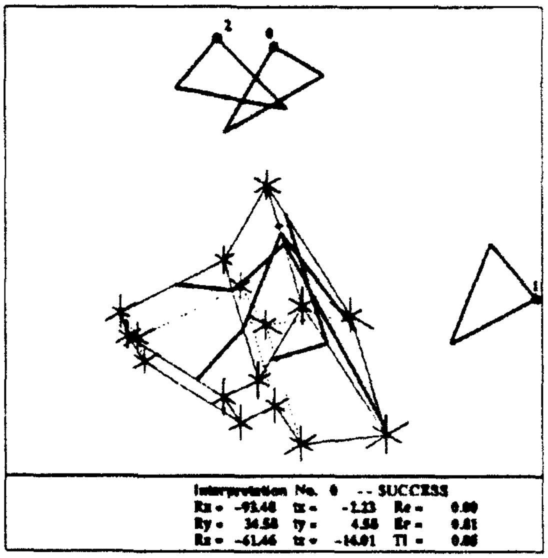
4.3 Absolute Accuracy

We estimated the absolute accuracy in pose determination with the sensor placement shown in Figure 10. The object is located with a known transformation (Case 1 ~ 6), and the object pose is estimated 10 times for each transformation. The mean and standard deviation of position errors (equation (12)) of each vertex are calculated. Table 2 shows the averages of the means and standard deviations of the position errors for all vertices. For all cases, the



```
Interpretation No. 0   -- SUCCESS
Rx = -93.48  tx =   -2.23  Re =   0.29
Ry =  34.58  ty =    4.58  Er =   0.92
Rz = -61.46  tz =  -14.01  Tl =   0.67
```

Figure 12: Experimented 3-D line segments and object recognition and position error estimation results for an arbitrary pose.



```
Interpretation No. 0   -- SUCCESS
Rx = -93.48  tx =   -2.23  Re =   0.00
Ry =  34.58  ty =    4.58  Er =   0.01
Rz = -61.46  tz =  -14.01  Tl =   0.06
```

Figure 13: Simulated 3-D line segments and object recognition and position error estimation results for the object pose shown in Figure 12.

Table 2: Absolute accuracy in pose determination. The object pose is estimated for the object with a known transformation. In all cases, $\omega = 0°$, $\varphi = 0°$, and t_z = -6.75 mm.

Transformation			Δv_x (mm)	Δv_y (mm)	Δv_z (mm)
Case 1	t_x = 5mm, t_y = -5mm $\kappa = 0°$	Mean	-0.46	-0.10	-0.04
		Std	0.52	0.30	0.15
Case 2	t_x = 5mm, t_y = 0mm $\kappa = 0°$	Mean	-0.35	-0.34	0.06
		Std	0.49	0.38	0.13
Case 3	t_x = 5mm, t_y = 0mm $\kappa = 30°$	Mean	-0.31	-0.39	0.24
		Std	0.20	0.14	0.07
Case 4	t_x = 0mm, t_y = -5mm $\kappa = 30°$	Mean	0.66	0.22	0.34
		Std	0.32	0.18	0.09
Case 5	t_x = 10mm, t_y = 0mm $\kappa = 60°$	Mean	-0.11	0.04	0.11
		Std	0.16	0.13	0.15
Case 6	t_x = 10mm, t_y = -5mm $\kappa = 60°$	Mean	-0.32	0.06	0.12
		Std	0.27	0.24	0.16

standard deviations of vertex position errors are within 0.6 mm. These values are consistent with the simulation results for the same transformations.[2]

4.4 Relative Accuracy

The relative accuracy in pose determination was estimated as follows.

1. The object is placed at an arbitrary pose in the workspace.
2. The object pose is estimated initially.
3. The object is moved in the x direction by 5mm.
4. The object pose is estimated again and compared with the initial pose.
5. Steps 3 and 4 are repeated.

Figure 14 shows the experimental results. The estimated x component of the translation changes linearly by 5mm and the y component is almost constant. The difference between

[2] In the simulation the standard deviations of vertex position errors are about 0.5 mm assuming the measurement error of the range finder to be σ = 0.3 mm.

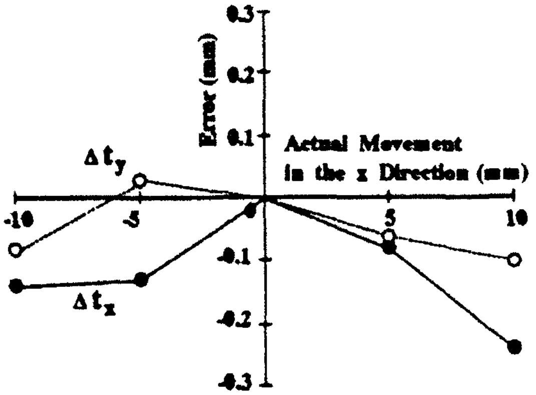


Figure 14: Relative accuracy in pose determination.

the actual and estimated translation components is within ± 0.25 mm. Similar experiments for 10 different initial object poses resulted in almost the same relative accuracy.

5 Conclusion

We have presented a method for estimating uncertainty in determining the pose of a polyhedral object when using multiple light-stripe range finders.

An object recognition method based on an interpretation tree search has been used to determine the object pose. In this method, 3-D line segments obtained by the range finders are consistently matched to model faces based on geometric constraints. We have introduced triplet constraints to dramatically speed pruning of the interpretation tree.

We have determined the relationship between uncertainty in object pose determination and sensing error. The pose error of an object can be estimated from the covariance matrix of the endpoint positions of sensed line segments.

Experiments with simple light-stripe range finders show that our method makes it possible to estimate how accurately the pose of an object can be determined.

Acknowledgment

The authors would like to thank Carlo Tomasi for useful discussions and comments on this work. Thanks to Jim Moody and Mark DeLouis who made the laser projectors and controller, and to Shin-ichi Yoshimura who helped with the experiments. John Bares and Nathan Fullerton read the manuscript carefully and greatly improved its readability. The authors also thank the members of the Vision and Autonomous Systems Center of Carnegie Mellon University for their valuable comments and suggestions.

Appendix

A Geometric Constraints

This appendix gives unary and binary constraints used in our object recognition method.

- Length constraint (Unary constraint)

  If a line segment lies on a model face, the length of the segment must be less than or equal to the maximum distance between two vertices of the model face.

- Distance constraint (Binary constraint)

  If two line segments lie on two different model faces, the range of distances between the two line segments must be within the range of distances between the corresponding two model faces.

- Adjacency constraint (Binary constraint)

  Two adjacent line segments from the same range finder must be assigned to two adjacent model faces.[3]

- *Intersection constraint (Binary constraint)*

  If two line segments which come from different range finders intersect each other, these line segments must be assigned to the same model face.

When a new node at the i th level is reached in the interpretation tree, a new pairing (S_i, M_{p_i}) is generated, which must be subjected to the unary constraints. Also $i-1$ new induced pairs of pairings, $[(S_i, M_{p_i}), (S_j, M_{p_j})]$ for $j = 1, \ldots, i-1$ must be subjected to the binary constraints.

B Triplet Constraints

In this appendix, we present two triplet constraints used in our object recognition method.

B.1 Surface Normal Constraint

Intersecting line segments can define the normal of the face on which the line segments lie. In Figure 15, let $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ denote the unit direction vectors of intersecting line segments

[3] Two model faces which share a vertex are regarded as adjacent.

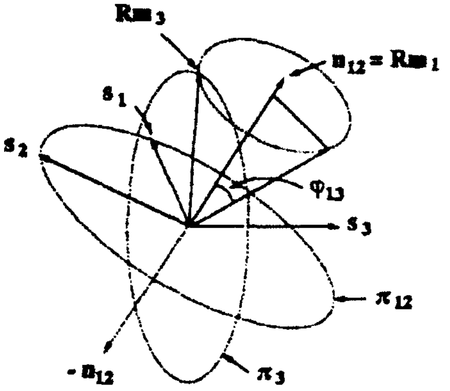


Figure 15: Surface normal constraint. For the constraint to be satisfied, the transformed surface normal $\mathbf{Rm}_3$ should be on the plane π_3 whose normal is $\mathbf{s}_3$ and also on the conical surface defined by the surface normal $\mathbf{n}_{12}$ and the angle φ_{13}.

S_1 and S_2 respectively. The unit normal of a plane π_{12}, on which those line segments lie, is represented by

$$\boldsymbol{n}_{12} = \frac{\boldsymbol{s}_1 \times \boldsymbol{s}_2}{\| \boldsymbol{s}_1 \times \boldsymbol{s}_2 \|}. \tag{13}$$

Let $\boldsymbol{m}_1$ be the unit normal of a model face, M_1, which is assigned to the two line segments, and let $\boldsymbol{R}$ denote the rotation component of the transformation from the model coordinate frame to the world coordinate frame. The unit normal of model face M_1 in the world coordinate frame, which is given by $\boldsymbol{Rm}_1$, is set to equal the unit normal $\boldsymbol{n}_{12}$ of the plane π_{12} or $-\boldsymbol{n}_{12}$. One direction is chosen such that the normal of the plane π_{12} is directed toward the range finders from which the line segments S_1 and S_2 were obtained.

Let S_3 denote another line segment which does not lie on the plane π_{12}. A possible model face M_3, matched to the line segment S_3, must satisfy the following conditions:

- The angle between the two model faces is invariant under a rigid transformation, that is, $\angle(\boldsymbol{Rm}_1, \boldsymbol{Rm}_3) = \angle(\boldsymbol{m}_1, \boldsymbol{m}_3) = \varphi_{13}$.

- The direction vector of the third line segment is perpendicular to the normal of the assigned model face, that is, $\boldsymbol{s}_3 \perp \boldsymbol{Rm}_3$.

Consequently, the unit normal $\boldsymbol{Rm}_3$ of the transformed model face M_3 can be obtained by solving the following equations simultaneously.

$$\begin{aligned} m_{1x}m_{3x} + m_{1y}m_{3y} + m_{1z}m_{3z} &= \cos\varphi_{13} \\ s_{3x}m_{3x} + s_{3y}m_{3y} + s_{3z}m_{3z} &= 0 \\ m_{3x}^2 + m_{3y}^2 + m_{3z}^2 &= 1 \end{aligned} \tag{14}$$

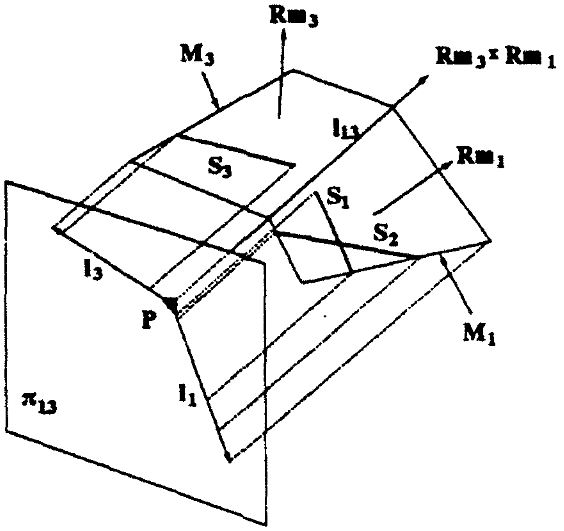


Figure 16: Projection constraint. The projected points of the endpoints of line segments S_1, S_2 and S_3 must be within the ranges l_1 and l_3 respectively.

where

$$\boldsymbol{Rm}_1 = \begin{pmatrix} m_{1x} \\ m_{1y} \\ m_{1z} \end{pmatrix}, \quad \boldsymbol{Rm}_3 = \begin{pmatrix} m_{3x} \\ m_{3y} \\ m_{3z} \end{pmatrix}, \quad \boldsymbol{s}_3 = \begin{pmatrix} s_{3x} \\ s_{3y} \\ s_{3z} \end{pmatrix}.$$

If no real root exists to these equations, the chosen triplet $[(S_1, M_1), (S_2, M_1), (S_3, M_3)]$ is inconsistent, and this interpretation is discarded. Since the surface normals $\boldsymbol{Rm}_1$ and $\boldsymbol{Rm}_3$ in Figure 15 correspond to two unit surface normals, $\boldsymbol{m}_1$ and $\boldsymbol{m}_3$, in the model coordinate frame, the rotation matrix $\boldsymbol{R}$ can be computed [9].

B.2 Projection Constraint

A triplet surviving from the surface normal constraint is subjected to another triplet constraint. Suppose that the surface normal of a plane π_{13} is defined by the vector product of the transformed normals $\boldsymbol{Rm}_1$ and $\boldsymbol{Rm}_3$ of two model faces M_1 and M_3 in Figure 16.[4] Let P denote the intersection point of the line l_{13} (the intersection line of the two transformed model faces) with the plane π_{13}. When any point on the transformed model face M_1 is projected onto the plane π_{13} along the direction of the line l_{13}, the projected point will be within the range denoted by l_1 on the plane π_{13}. Similarly, the projection of any point on the transformed model face M_3 will fall on the range l_3. Since the three line segments S_1, S_2 and S_3 are on the transformed model faces M_1, M_1 and M_3 respectively, the projected points of

[4] The two model faces M_1 and M_3 are not necessarily adjacent.

the endpoints of the line segments must fall within the corresponding ranges. If one or more endpoints are out of the ranges, the triplet $[(S_1, M_1), (S_2, M_1), (S_3, M_3)]$ is inconsistent.

References

[1] P. J. Besl. Active, optical range imaging sensors. *Machine Vision and Applications*, 1:127–152, 1988.

[2] R. C. Bolles and P. Horaud. 3DPO: A three-dimensional part orientation system. *The International Journal of Robotics Research*, 5(3):3–26, 1986.

[3] H. H. Chen. Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):530–541, June 1991.

[4] R. E. Ellis. Geometric uncertainties in polyhedral object recognition. *IEEE Transactions on Robotics and Automation*, 7(3):361–371, June 1991.

[5] T. J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-D objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1140–1157, November 1989.

[6] O. D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *The International Journal of Robotics Research*, 5(3):27–52, 1986.

[7] P. J. Flynn and A. K. Jain. BONSAI : 3-D object recognition using constrained search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1066–1075, October 1991.

[8] S. J. Gordon and W. P. Seering. Real-time part position sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):374–386, May 1988.

[9] W. E. L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *The International Journal of Robotics Research*, 3(3):3–35, 1984.

[10] W. E. L. Grimson. Sensing strategies for disambiguating among multiple objects in known poses. *IEEE Journal of Robotics and Automation*, RA-2(4):196–213, December 1986.

[11] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. The MIT Press, Cambridge, MA, 1990.

[12] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, July 1988.

[13] K. Ikeuchi and T. Kanade. Modeling sensors: Toward automatic generation of object recognition program. *Computer Vision, Graphics, and Image Processing*, 48:50–79, 1989.

[14] D. W. Murray and D. B. Cook. Using the orientation of fragmentary 3D edge segments for polyhedral object recognition. *International Journal of Computer Vision*, 2:153–169, 1988.

[15] D. W. Murray. Model-based recognition using 3D shape alone. *Computer Vision, Graphics, and Image Processing*, 40:250–266, 1987.

[16] M. Oshima and Y. Shirai. Object recognition using three-dimensional information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(4):353–361, 1983.

[17] K. C. Sahoo and C. H. Menq. Localization of 3-D objects having complex sculptured surfaces using tactile sensing and surface description. *Journal of Engineering for Industry*, 113:85–92, February 1991.

[18] S. Shekhar, O. Khatib, and M. Shimojo. Object localization with multiple sensors. *The International Journal of Robotics Research*, 7(6):34-44, December 1988.