[13] N. Grcywacz, J. Smith, and A. Yuille, "A common theoretical framework for visual motion's spatial and temporal coherence," CH2716, pp. 148–155, 1989.
[14] R. Shepard and L. Cooper, *Mental Images and their Transformations*. Cambridge, MA: MIT Press, 1982.
[15] M. J. Black and P. Anandan, "Robust dynamic motion estimation over time," in *Proc. Computer Vision and Pattern Recognition*, Maui, Hawaii, 1991, pp. 296–302.
[16] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-14, no. 9, pp. 910–927, Sept. 1992.
[17] , "Multigrid Bayesian estimation of image motion fields using stochastic relaxation," in *Proc. Second Int. Conf. Computer Vision*, Tampa, FL, 1988, pp. 354–362.
[18] ——, "Comparison of stochastic and deterministic solution methods in Bayesian estimation of 2D motion fields" *Image and Vision Computing*, vol. 9, no. 4, pp. 215–228, Aug. 1991.
[19] J. Hutchinson, C. Koch, J. Luo, and C. Mead, "Computing motion using analog and binary resistive network," *IEEE Computer*, vol. 21, pp. 52–63, Mar. 1988.
[20] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 6, pp. 721–741, Nov. 1987.
[21] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, vol. 1. New York: Interscience, 1970.
[22] D. J. Heeger, "Optical flow from spatiotemporal filters," in *Proc. First Int. Conf. Computer Vision*, London, England, 1987.
[23] I. K. Sethi and R. C. Jain, "Finding trajectories of feature points in a monocular image sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 1, Jan. 1987.
[24] K. Rangarajan and M. Shah. "Establishing motion correspondence," in *Proc. Computer Vision and Pattern Recognition*, Lahaina, Hawaii, 1991, pp. 103–108.
[25] K. Chaudhury and R. Mehrotra, "Optical flow estimation using smoothness of intensity trajectories," CVGIP: *Image Understanding*, vol. 60, no. 2, pp. 230–244, Sept. 1994.
[26] H. H. Nagel, "Displacement vectors derived from second-order intensity variation in image sequences," *Computer Vision Graphics and Image Processing*, vol. 21, pp. 85–117, 1983.
[27] ——, "On the estimation of optical flow: Relations between different approaches and some new results," *Artificial Intell.*, vol. 33, pp. 299–324, 1987.
[28] E. C. Hildreth, "Computations underlying the measurement of visual motion," *Artificial Intell.*, vol. 23, pp. 309–354, 1984.
[29] D. Marr, *Vision*. New York: Freeman, 1982.
[30] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt, "Performance of optical flow techniques," in *Proc. Computer Vision and Pattern Recognition*, Champaign, IL, 1992, pp. 236–242.

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

Keiichi Kemmotsu and Takeo Kanade

Abstract— **The pose (position and orientation) of a polyhedral object can be determined by using a set of simple light-stripe range sensors. Data from these sensors, however, inherently contains errors which result in pose uncertainty. Therefore, in addition to calculating the pose itself, it is often desirable to estimate how certain the pose determination is. This paper presents a method for estimating the uncertainty in determining the pose of an arbitrarily positioned object with three light-stripe range finders. We analyze the perturbation of a least squares fit of sensed three-dimensional (3-D) segments to object faces, and obtain a relationship between the sensing error and the object pose error. Experiments demonstrate that the method provides the estimate of accuracy in pose determination.**

I. 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 pose of the object. 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 the pose uncertainty. Using an error analysis based on the convergence properties of the least squares fit, we obtain a relationship between the covariance matrix of the endpoint positions of line segments and the covariance matrix of the positions of object vertices. The pose uncertainty can then be estimated from this relationship.

Manuscript received August 20, 1993; revised June 22, 1994. This work was supported by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, under Contract F33615-90-C-1465, ARPA Order 7597-597. 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. Portions of this paper were presented at the IEEE International Conference on Robotics and Automation, Atlanta, GA, 1993.

K. Kemmotsu was with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with the Advanced Technology Research Center, Mitsubishi Heavy Industries, Ltd., Kanazawa-ku, Yokohama 236 Japan.

T. Kanade is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

IEEE Log Number 9411924.

A. 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 [1] and some range imaging techniques are very fast [2], 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 and 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 pose of objects is *a priori* known. Gordon and Seering [6] showed that object pose can be determined precisely with one simple light-stripe range finder providing that the pose of the object is known *a priori* 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 correspondences between sensed features and model features. To find the correspondences, an interpretation tree search method with geometric constraints is used. Grimson and Lozano-Pérez [7] demonstrated that local unary and binary geometric constraints are very effective in reducing the size of an interpretation tree. 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 [5], [10]. Grimson [8] 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. Since the pose uncertainty of an object can be represented by the covariance matrix of the position of each vertex, we explore a pose uncertainty estimation method that uses the covariance matrix of the endpoint positions of sensed line segments.

II. 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 object recognition example. A simple light-stripe range finder projects a light plane onto the faces of an object and measures 3-D line segments created by the light-stripe as shown in Fig. 1. Three identical range finders are placed in the world coordinate frame as shown in Fig. 2. The range finders obtain 3-D line segments as shown in Fig. 3. The object's pose is successfully determined as shown in Fig. 4. In this section, we describe our object recognition and pose determination technique.

A. Interpretation Tree Search by Geometric Constraints

The interpretation tree search technique with local unary and binary geometric constraints finds a consistent set of pairings (S_1, M_{p_1}), (S_2, M_{p_2}), $\cdots$, (S_k, M_{p_k}), where M_{p_i} is a model face which corresponds to line segment S_i. The unary constraints check the consistency of a segment-face pairing and the binary constraints check the consistency of two pairings.

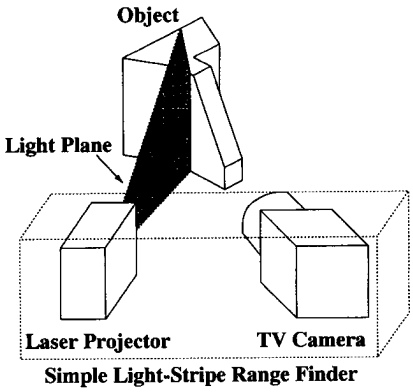

Fig. 1. A simple light-stripe range finder and an object model.

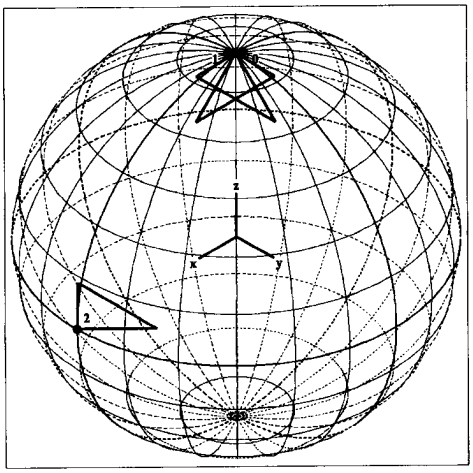

Fig. 2. Sensor placement for object recognition. Sensors 0 and 1 are placed on the z axis, directed toward the origin. Their light planes, which are displayed as triangles, are orthogonal. Sensor 2 is placed on the x axis and its light plane lies on the x-y plane.

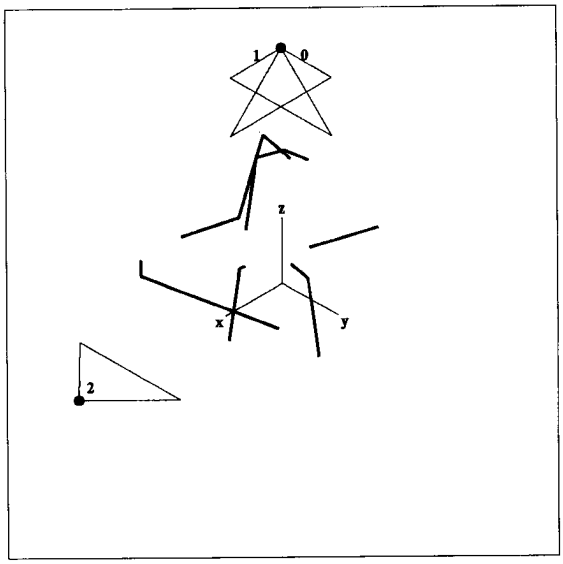

Fig. 3. Obtained 3-D line segments on object faces.

Our unary and binary constraints for segment-face matching are weaker than those for face-face and edge-edge matching in Grimson's work [9] since line segments carry less information than faces and edges. Therefore, after applying the unary and binary constraints, we apply triplet constraints which check three segment-face pairings

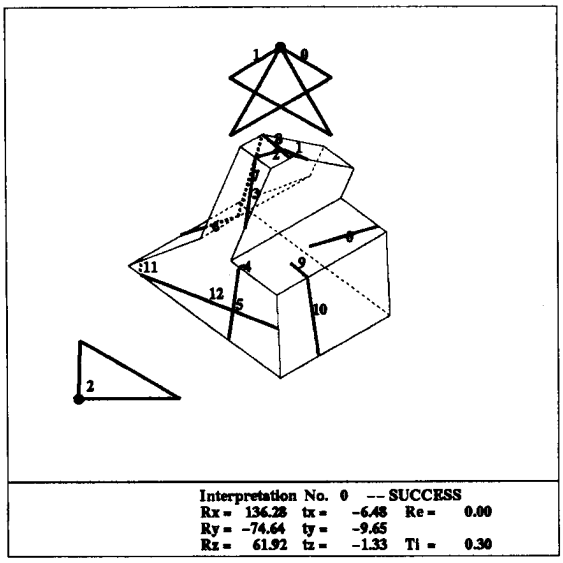

Fig. 4. An object recognition result. Estimated transformations $\omega(R_x)$, $\varphi(R_y)$, and $\kappa(R_z)$ are given in degrees and t_x, t_y, and t_z are given in mm. R_e is the standard deviation of the distances between the endpoints of the line segments and the corresponding model faces. T_i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

to prune the interpretation tree more efficiently. 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.[1] The intersecting line segments can be used to calculate 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 the Appendix.

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 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 inconsistent segment-face pairings. Intersecting line segments should therefore be used as early as possible to prune the interpretation tree and save computation time.

B. Computing Transformations

Next, we solve for the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ of the transformation which maps points in the model coordinate frame into the world coordinate frame in such a manner that each line segment lies on the corresponding model face. 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}. 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

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

[1] In case that the intersection point of two line segments is near or at an endpoint of both lines, the line segments could belong to two adjacent faces, respectively, and they are not regarded as intersecting in our algorithm.

where $\boldsymbol{N}_{p_i}$ and D_{p_i} are the unit normal and offset of the model face M_{p_i}, respectively. The rotation and translation components are 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 $i = 1, \cdots, k$

$$E = \sum_{i=1}^{k} \int_{\boldsymbol{b}_i}^{\boldsymbol{e}_i} (\Delta d_i)^2 ds_i \tag{3}$$

where ds_i is an element of line segment S_i.

An initial rotation component for minimization is obtained by using a geometric relationship among three segment-face pairings which include intersecting line segments. In the event that the three pairings do not include intersecting line segments, a numerical polynomial-based technique is used to obtain a rotation component. 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. Unfortunately, these polynomial-based methods are very sensitive to noise and are also computationally expensive since an eighth-degree equation must be solved. On the other hand, the method which uses intersecting line segments is very fast and robust since a rotation component is obtained by solving a quadratic equation in the triplet constraint check. An initial translation component is computed by a least squares method.

C. Simulation

We run simulation to test the effectiveness of our object recognition method. We use a polyhedral object as shown in Fig. 1. Three hypothetical light-stripe range finders are placed in the world coordinate frame as shown in Fig. 2. The object is then randomly located in the world coordinate frame. 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. We obtain feasible interpretations 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 I. 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 s 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. For example, the intersecting line segments no. 5 and no. 12 in Fig. 4 play a crucial role to decrease the number of nodes of the interpretation tree and the computation time is decreased from 20 to 0.3 s. 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 s.

III. Geometric Uncertainty 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 estimate of the object. This section describes our technique for estimating the pose uncertainty.

TABLE I
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

A. Estimating Pose Uncertainty

The transformation error is defined as a perturbation around the correct transformation with respect to the sensing error. We denote transformation variables by

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

where t_x, t_y, and t_z are translation components and ω, φ, and κ are rotation angles around x, y, and z axes in the world coordinate frame, respectively.[2] Let

$$\boldsymbol{s} = (x_1, y_1, z_1, x_2, y_2, z_2, \cdots, x_{2k}, y_{2k}, z_{2k})^T$$

be a vector of endpoint pairs $(x_{2i-1}, y_{2i-1}, z_{2i-1})$ and (x_{2i}, y_{2i}, z_{2i}) of line segments S_i for $i = 1, \cdots, k$. The pose of an object is determined by minimizing the residual E of (3) with respect to $\boldsymbol{\xi}$. The necessary condition for E to reach an extremum is given as

$$\frac{\partial E}{\partial t_x} = \frac{\partial E}{\partial t_y} = \frac{\partial E}{\partial t_z} = \frac{\partial E}{\partial \omega} = \frac{\partial E}{\partial \varphi} = \frac{\partial E}{\partial \kappa} = 0. \tag{4}$$

Now to examine the transformation error $\Delta\boldsymbol{\xi}$ caused by the sensing error $\Delta\boldsymbol{s}$, we linearize these nonlinear equations around the approximate solution which corresponds to the correct transformation and endpoints,

$$\boldsymbol{A}\Delta\boldsymbol{\xi} \cong -\boldsymbol{B}\Delta\boldsymbol{s} \tag{5}$$

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

Furthermore, a relationship between the transformation error $\Delta\boldsymbol{\xi}$ and the position error $\Delta\boldsymbol{v}_j$ of a vertex $\boldsymbol{v}_j$ is given by

$$\Delta\boldsymbol{v}_j \cong \boldsymbol{D}_j\Delta\boldsymbol{\xi} \tag{6}$$

where $\boldsymbol{D}_j$ is the Jacobian matrix of $\boldsymbol{v}_j$ with respect to $\boldsymbol{\xi}$. By substituting (5) into (6), the relationship between the sensing error and the position error becomes

$$\Delta\boldsymbol{v}_j \cong -\boldsymbol{D}_j\boldsymbol{A}^{-1}\boldsymbol{B}\Delta\boldsymbol{s}. \tag{7}$$

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

$$\begin{aligned}\boldsymbol{C}_{v_j} &\equiv \boldsymbol{E}(\Delta\boldsymbol{v}_j\Delta\boldsymbol{v}_j^T) \\ &= \boldsymbol{D}_j(\boldsymbol{A}^{-1}\boldsymbol{B})\boldsymbol{C}_s(\boldsymbol{A}^{-1}\boldsymbol{B})^T\boldsymbol{D}_j^T\end{aligned} \tag{8}$$

[2]While there are many representations for rigid rotations, we choose Euler angles since the representation consists of only three parameters with no mutual constraints.

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

$$(\Delta v_{j_x}, \Delta v_{j_y}, \Delta v_{j_z}) = (\sqrt{C_{v_j 11}}, \sqrt{C_{v_j 22}}, \sqrt{C_{v_j 33}}). \tag{9}$$

B. Examples

The following are some examples of estimating geometric uncertainty in pose determination. Given the shape of an object, the object's pose in world coordinates, and a sensor 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 with standard deviation of 1 mm).[3] We further assume that any two endpoints are independently measured and that their respective errors are not related (though the mechanism of the sensing error of a range finder is complex in practice [11]). Thus, the covariance matrix $\boldsymbol{C}_s$ of the line segments' endpoint positions becomes the identity matrix. We can estimate the uncertainty of each vertex of the object with (8).

Given a model as shown in Fig. 1, a sensor placement as in Fig. 2, and the same transformation as in Fig. 4, the estimated pose uncertainty is shown in Fig. 5. In this figure, the lengths of three bars on each vertex along x, y, and z directions are given by (9), and show the uncertainty associated with the position of each vertex. Another example with a different object pose is shown in Fig. 6. Fig. 7 shows the object in the same pose as Fig. 6, but under a different sensor placement. The position error in Fig. 7 is much larger than that in Fig. 6. Uncertainty on vertices of an object depends on the object's pose with respect to the range finders, that is, the number of line segments, the spatial distribution of the line segments, the number of object faces on which the line segments fall, and the spatial distribution of the faces.[4]

IV. EXPERIMENTS

This section presents experimental results of recognizing an object and estimating pose uncertainty. Each light-stripe range finder is composed of a TV camera with a 16-mm 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 Fig. 8. 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 0.3 mm (standard deviation) within the workspace.[5]

A. Experimental Results

An object like the one depicted in Fig. 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 Fig. 9. Fig. 10 shows obtained 3-D line segments and object recognition and pose uncertainty estimation results. For comparison,

[3]The noise is approximately three times as large as the measurement error of the light-stripe range finder in the experiments (see Section IV).

[4]The object's shape as shown in these figures is just arbitrary. Pose uncertainty must be also related with the shape of an object. However, quantitative analysis for the relationship is beyond the scope of the current paper.

[5]The coefficients of each light plane equation and the coefficients of the projective transformation between the light plane and the image plane of each TV camera are calibrated by using four reference points in the same world coordinate frame.

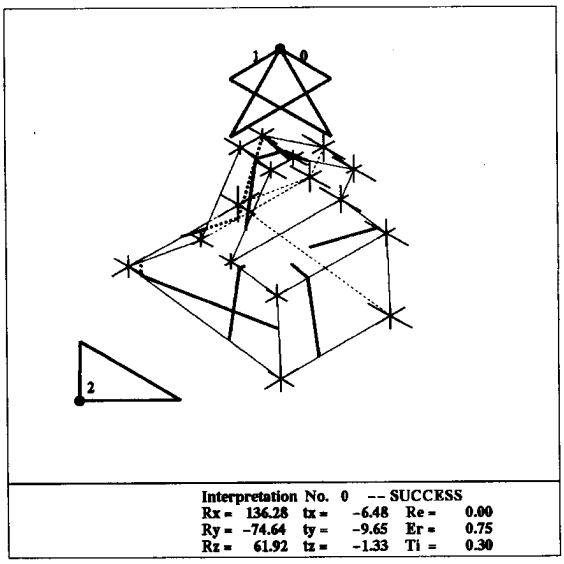

Fig. 5. A pose uncertainty estimation result after recognizing the object. Three triangles show the light planes of three range finders, and thick solid lines show 3-D line segments obtained from the range finders. Three bars(thin solid line) on each vertex along x, y, and z directions show the uncertainty in pose determination. For display purpose, those lengths equal $12\Delta v_{j_x}$, $12\Delta v_{j_y}$, and $12\Delta v_{j_z}$, respectively. E_r(mm) is the average position error of all vertices.

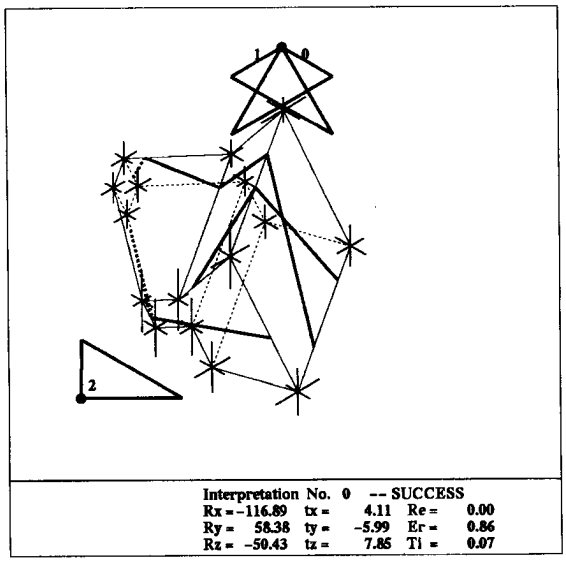

Fig. 6. A pose uncertainty estimation result under the same sensor placement as in Fig. 5 but with a different object pose.

Fig. 11 shows a simulation result with the same object pose under the same sensor placement as the experiment shown in Fig. 10. The recognition time in the experiment is 0.67 s, while only 0.05 s 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 visited nodes. Note that the line segments no. 0 and no. 1 and the line segments no. 6 and no. 7 in Fig. 10 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.

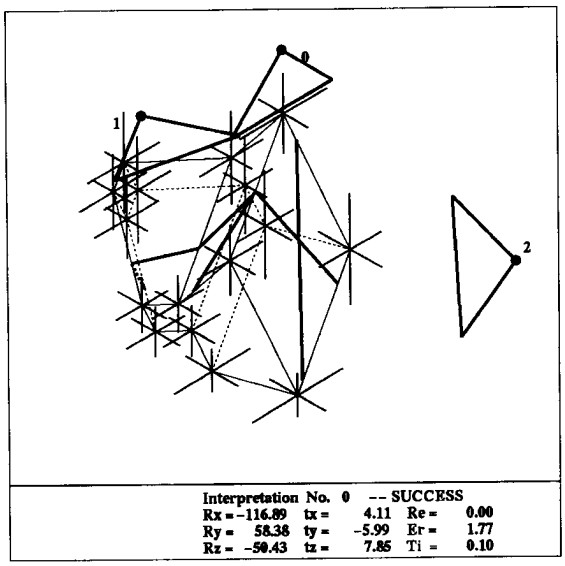

Fig. 7. A pose uncertainty estimation result with the same object pose as in Fig. 6 but under a different sensor placement. Longer bars on vertices show that position errors on the vertices are much larger in their directions.

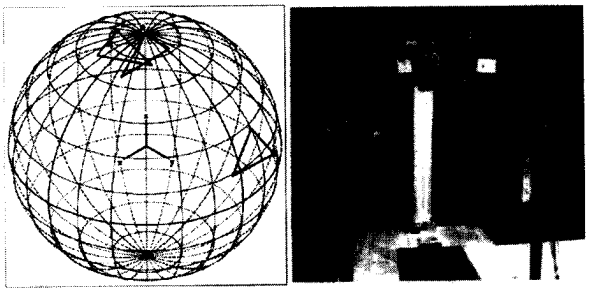

Fig. 8. Sensor placement for experiments.

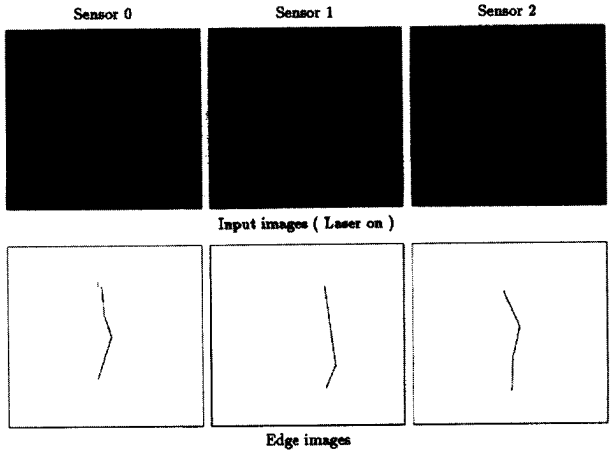

Fig. 9. Input images for the three light-stripe range finders and extracted edge images. The object is placed on a support cube whose size is 60 × 60 × 60 mm. The cube is not regarded as a part of the object.

We tried similar experiments with several different poses. A few 3-D line segments were occluded in some experimental results, while the line segments appeared on object faces in corresponding simulation results. This is because the range finder simulator regards the light source and the viewpoint as the same point. Throughout the trials, the experimental results are consistent with the simulation results except for recognition time and occlusion.

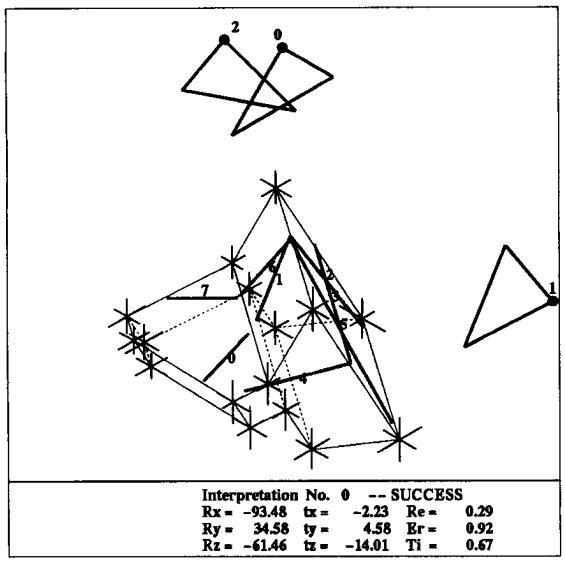

Fig. 10. Experimented 3-D line segments and object recognition and pose uncertainty estimation results for an arbitrary pose.

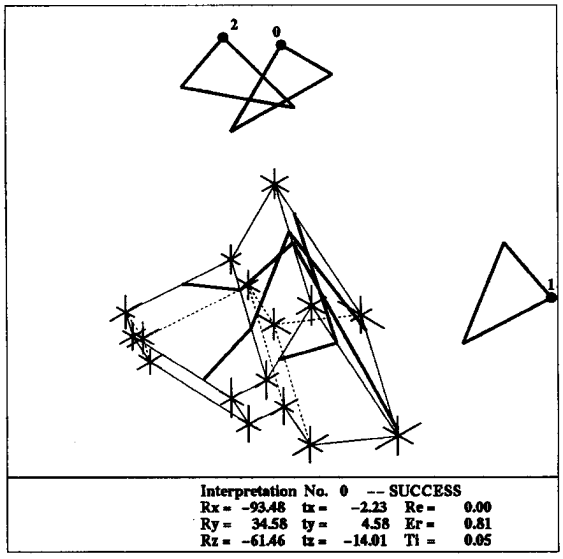

Fig. 11. Simulated 3-D line segments and object recognition and pose uncertainty estimation results for the object's pose shown in Fig. 10.

B. Absolute Accuracy

We estimated the absolute accuracy in pose determination under the sensor placement shown in Fig. 8. The object is located with a known transformation (Case 1 $\sim$ 6), and the object pose is estimated 10 times for each transformation. The mean and standard deviation of position errors (9) of each vertex are calculated. Table II shows the averages of the means and standard deviations of the position errors for all vertices. For all cases, the standard deviations of vertex position errors are within 0.6 mm. These values are consistent with the simulation results for the same transformations; 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.

C. Relative Accuracy

The relative accuracy in pose determination was estimated as follows.

TABLE II
ABSOLUTE ACCURACY IN POSE DETERMINATION

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

The pose and vertex positions are estimated 10 times for the object with each transformation. In all cases, $\omega = 0°$, $\varphi = 0°$, and t_z = -6.75 mm. The mean and standard deviation of position errors Δv_{j_x}, Δv_{j_y}, and Δv_{j_z} for each vertex are taken. The "Mean" and "Std" values in the table are their averages over all vertices.

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 5 mm.
4) The object pose is estimated again and compared with the initial pose.
5) Steps 3 and 4 are repeated.

Fig. 12 shows the experimental results. The estimated x component of the translation changes linearly by 5 mm and the y component is almost constant. The difference between the actual and estimated translation components is within ± 0.25 mm. The difference when moving the object in the y direction is also within ± 0.25 mm. Similar experiments for 10 different initial object poses resulted in almost the same relative accuracy.

As for relative accuracy in rotation angles, the object poses with three different rotation angles ($\kappa = 0°$, 30°, and 60°) were estimated. The difference between the actual and estimated rotation angles is within $\pm 1.0°$. The transformation error would not be very asymmetric since three range finders were placed viewing the object from different perspectives in the experiments. We confirmed through simulation that the accuracy for three translation components and three rotation angles with the same object pose under the same sensor placement as shown in Fig. 11 behaved in almost the same way, respectively. Pose uncertainty, however, depends on the geometric relationship between a sensor placement and an object pose as we mentioned in Section III.

V. 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

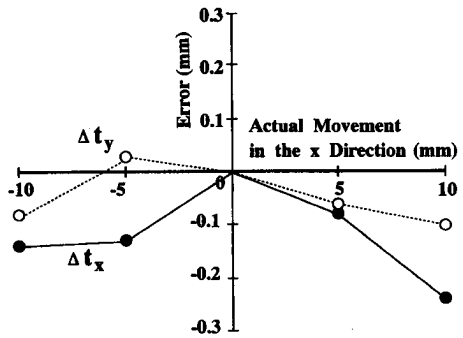

Fig. 12. Relative accuracy in pose determination.

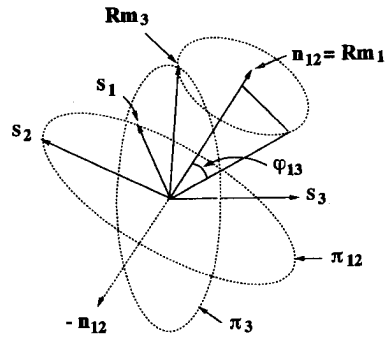

Fig. 13. Triplet constraints. For the constraints to be satisfied, the transformed surface normal $\boldsymbol{Rm}_3$ should be on the plane π_3 whose normal is $\boldsymbol{s}_3$ and also on the conical surface defined by the surface normal $\boldsymbol{n}_{12}$ and the angle φ_{13}.

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 sensing error and pose uncertainty. The pose uncertainty can be estimated from the covariance matrix of the endpoint positions of sensed line segments. Experiments with three simple light-stripe range finders show that our method makes it possible to estimate how accurately the pose of an object can be determined.

The problem of designing an optimal sensor placement of three light-stripe range finders with which to determine the pose of an object is currently being addressed [12]. Our object recognition and pose uncertainty estimation techniques are used to evaluate the goodness of a sensor placement. Other important future directions are to investigate sensor models which exactly reflect sensing errors of a light-stripe range finder and to extend our techniques to determine the pose of an object in cluttered scenes.

APPENDIX

In the appendix, we present the triplet constraints used in our object recognition method. Intersecting line segments can define the normal of the face on which the line segments lie. In Fig. 13, let $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ denote the unit direction vectors of intersecting line segments S_1 and S_2, respectively. The unit normal of a plane π_{12}, on which those line segments lie, is represented by

$$\boldsymbol{n}_{12} = \frac{\boldsymbol{s}_1 \times \boldsymbol{s}_2}{\|\boldsymbol{s}_1 \times \boldsymbol{s}_2\|}. \tag{10}$$

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 a quadratic equation. If no real root exists to the equation, 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 Fig. 13 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 [7].

ACKNOWLEDGMENT

The authors would like to thank C. Tomasi for useful discussions and comments on this work. They would also like to thank J. Moody and M. DeLouis who made the laser projectors and controller, and to S. Yoshimura who helped with the experiments. J. Bares and N. 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 and the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] P. J. Besl and R. C. Jain, "Three-dimensional object recognition," *Computing Surveys*, vol. 17, no. 1, pp. 75–145, Mar. 1985.
[2] P. J. Besl, "Active, optical range imaging sensors," *Mach. Vision Appl.*, vol. 1, pp. 127–152, 1988.
[3] H. H. Chen, "Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 13, no. 6, pp. 530–541, June 1991.
[4] R. E. Ellis, "Geometric uncertainties in polyhedral object recognition," *IEEE Trans. Robot. Automat.*, vol. 7, no. 3, pp. 361–371, June 1991.
[5] O. D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 27–52, 1986.
[6] S. J. Gordon and W. P. Seering, "Real-time part position sensing," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 10, no. 3, pp. 374–386, May 1988.
[7] W. E. L. Grimson and T. Lozano-Pérez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robot. Res.*, vol. 3, no. 3, pp. 3–35, 1984.
[8] W. E. L. Grimson, "Sensing strategies for disambiguating among multiple objects in known poses," *IEEE J. Robot. Automat.*, vol. RA-2, no. 4, pp. 196–213, Dec. 1986.
[9] W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*. Cambridge, MA: MIT Press, 1990.
[10] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *J. Opt. Soc. Am. A*, vol. 5, no. 7, pp. 1127–1135, July 1988.
[11] K. Ikeuchi and T. Kanade, "Modeling sensors: Toward automatic generation of object recognition program," *Comp. Vision, Graphics, and Image Process.*, vol. 48, pp. 50–79, 1989.
[12] K. Kemmotsu and T. Kanade. "Sensor placement design for object pose determination with three light-stripe range finders," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 1994, pp. 1357–1364.