Published in the Proceedings of the 1995 IEEE Conference on Robotics and Automation

Object Recognition and Localization from Scanning Beam Sensors

Aaron S. Wallack * John F. Canny †
Computer Science Division
University of California, Berkeley

Abstract

Model based object recognition and object localization are fundamental problems in industrial automation. We present techniques which use a scanner composed of binary light beam sensors to quickly recognize objects (as fast as 5 microseconds) and to accurately localize objects (0.025 millimeters), and we describe localization and recognition experiments and present results. Binary sensors only sense whether the part is present or absent at a particular location, and their high performance is due to their simple specification. Fast recognition is achieved by using indexing to solve the correspondence problem, the problem of interpreting the sensed data as model features; indexing involves using the sensed data to directly look up the correspondence information using a precomputed indexing table. Since each experiment only produces a single indexing vector, indexing tables need to be complete; in this paper we detail a complete indexing construction method for flat polygonal and polyhedral objects.

1 Introduction

Model based object recognition and model based object localization are two fundamental problems in machine vision, and, consequently, industrial automation. Model based recognition deals with identifying a model O given a set of models $\{O_1, O_2, \ldots, O_n\}$; model based object localization deals with estimating an object's pose from an object model and sensed data. In this paper, we discuss efficient techniques for solving both these problems for scanning sensor data.

Scanning sensor data can be produced by moving an object relative to a scanning sensor and the positions where a sensor's output changes are termed scanline endpoints (refer Figure 1). Given the scanline endpoints and a set of modeled objects, the task is to identify the scanned object and estimate its pose (refer Figure 2). What makes this problem difficult is that scanning sensors perceive only a few boundary points from the object's silhouette. This problem is solvable because of the sensor's high precision (0.025 mm). In addition, we assume that objects are either flat convex or non-convex polygons, or convex or non-convex polyhedra stably resting on a horizontal surface; these assumptions reduce the system to finite number of two-dimensional problems.

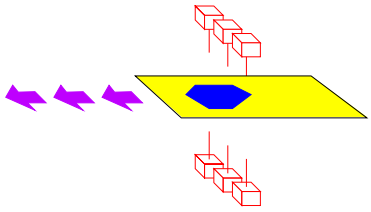

Figure 1: Scanning procedure

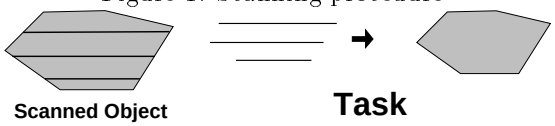

Figure 2: Given the scanline endpoints produced by scanning an arbitrary object, we determine the object's identity and pose

At the heart of both the recognition problem and the localization problem is the correspondence problem: interpreting the sensed data as model features. We utilize an indexing approach to solve the correspondence problem in constant time (as fast as five microseconds). We then validate each candidate interpretation via localization (if the object can be localized with the data, then it matches the data). Localization is performed in ten milliseconds and involves algebraic and numerical methods; the scanning sensor data is highly accurate and pose estimates are repeatable to within 0.025 mm and 0.1^o [12]. For comparison, commerical machine vision systems recognize and localize objects within an image via template matching to sub-pixel accuracies in sixty milliseconds.

Performance is the main reason for using scanning beam sensors; other considerations include ro-

*Supported by Fannie and John Hertz Fellowship and NSF FD93-19412
†Supported by NSF FD93-19412

bustness, speed, and monetary cost. Commercially available binary beam sensors are repeatable to less than 0.025 mm. In addition, binary beam sensors are robust, inexpensive, and insensitive to ambient lighting. Other sensors and techniques also follow the scanning paradigm: interpreting scanline camera data, interpreting continuous output from three dimensional range sensors, and registering objects from contact probe data.

This work is in the area of RISC (Reduced Intricacy Sensing and Control) robotics [1] which favors specialized, rather than general approaches. In a structured environment, such as an industrial workcell, specialized techniques can be faster, more efficient, and more robust than general techniques; furthermore, simpler sensors and control techniques are easy to setup and calibrate.

1.1 Previous Work

Most of the previous work in recognition, localization, and correspondence have been done in the context of machine vision. The problem of interpreting scanning sensor data can be handled by general machine vision techniques [8, 10, 7, 2, 5, 3], with the caveat that general approaches would be slower and less robust than our tailored approach.

In the machine vision literature, there are two main approaches to solving the correspondence problem, online combinatoric searches, and algorithms which utilize offline preprocessing.

The correspondence problem can be solved completely online using exhaustive brute force methods; exhaustive methods involve generating possible object poses by matching all possible tuples of sensed features to all possible tuples of model features, and then looking for find a consistent interpretation of the other sensed features. The problem with exhaustive approaches is that, without an effective pruning heuristic, the running time is on the order $O(n^k m^k)$ where n is the number of sensed features, m is the number of model features, and k is the tuple size. Huttenlocher and Ullman developed the alignment method in which the object's pose is determined using a minimal set of features, followed by verifying each hypothesized pose with the other sensed features [7]. Interpretation trees are a different approach which involves recasting the problem of finding a consistent interpretation into the problem of searching a tree where incompatible matches/interpretations are implicitly pruned [4, 6].

Improved performance can be achieved via offline preprocessing. Indexing techniques involve extracting indexing coordinates from tuples of sensed features, and discretizing these coordinates to index a table entry containing the correspondence information [2, 9, 8, 5]. The main advantage of indexing is that the indexing tables for all model feature tuples can be combined to form a single indexing table with the advantage that a single lookup compares a sensed feature tuple to all of the model feature tuples; in addition, the majority of the computations are performed offline. Unfortunately, previous methods for constructing indexing tables did not guarantee construction of complete tables, except in degenerate cases. Completeness is a critical issue for sparse sensing applications because each experiment may only provide a single indexing vector. We implemented the complete indexing table construction method described by Wallack and Canny [11]. Complete indexing tables also enable us to estimate the probability that the sensor will not be able to uniquely identify an object from a set of possible objects; this can be done by checking for overlap between the various indexing tables.

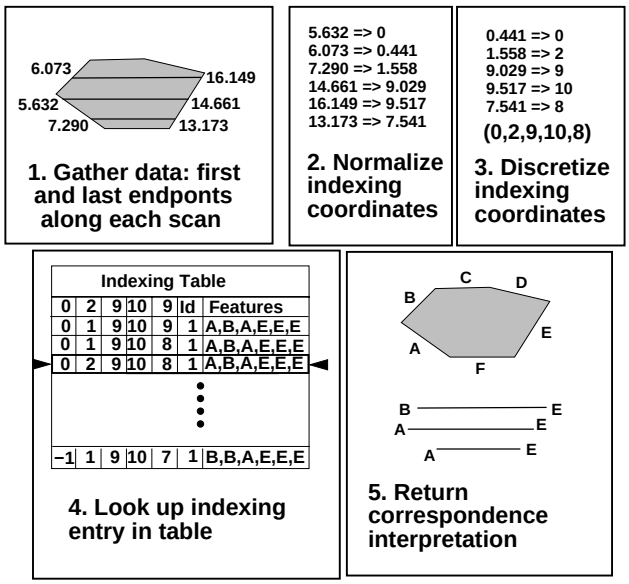

Figure 3: Indexing techniques involve discretizing the sensed data to index the entry containing valid interpretations

1.2 Indexing Overview

In this paper, we detail an indexing approach for solving the correspondence problem given scanline data. Figure 3 depicts the entire indexing process: The data is extracted and discretized, and then the discretized data is used to index into a table entry containing all valid feature correspondences.

The key idea of Wallack and Canny's indexing table construction algorithm [11] is that each indexing table entry corresponds to a finite number of continuous regions in configuration space, and these regions

are separated by only two types of boundaries: discretization boundaries separating configurations discretized to different indexing coordinates, and correspondence boundaries separating configurations with different correspondences information. We can enumerate all of the indexing table entries by enumerating all of these regions in configuration space; one way to enumerate all of these regions is to enumerate all of the pairwise intersections of the boundary curves.

1.3 Outline

The rest of the paper is organized in the following manner. In section 2 we describe the theoretical framework. In section 3 we detail the complete construction of the indexing table. The pose estimation technique is discussed in section 4. In section 5 we highlight the performance of scanning sensing with experimental data, and conclude by summarizing the results and advantages of these techniques.

2 Theoretical Framework

In this section, we discuss geometric foundation, the normalization procedure, and pose parameterization.

2.1 Geometric Foundation

Scanning sensors perceive only a few points from a silhouette of an object (a projection along the axis of the scanning beams). Since the object is assumed to be either a flat polygon or a polyhedron stably resting on a horizontal surface, we only need to consider a finite number of two-dimensional polygonal silhouettes. We make some additional restrictions:

1. Parts are singulated (presented one at a time).
2. Parts are not occluded by the environment.
3. Objects are detectable by scanning sensors.

2.2 Indexing Extremal Scanline Endpoints

To achieve a fixed number of indexing coordinates, we index using only the first and last scanline endpoints for each scanning sensor. The internal scanline endpoints, due to either non-convex objects, or internal holes, improve the localization estimate. The model features corresponding to the internal scanline endpoints are determined after localizing the object using only the extremal scanline endpoints.

2.3 Normalization

Normalization is a technique for reducing the effective number of degrees of freedom in order to condense the indexing table; normalization involves extracting correspondence-dependent information from the sensed data, and disregarding correspondence-independent information. Scanline endpoints are *normalized* in order to remove correspondence-irrelevant information. The object's position along the scanning path is irrelevant to interpreting the data, only the relative positions of the scanline endpoints are correspondence-relevant. Scanline endpoints are normalized by subtracting out the position of a particular scanline endpoint, the reference scanline endpoint (refer Figure 4). The term *normalized scanline endpoint* ($\overline{x_i}$) refers to the position of a scanline endpoint relative to the reference scanline endpoint. k scanning beam sensors generate $2k-1$ normalized scanline endpoints. Since the median sensor will register the object when the object is registered by at least half of the sensors, the first scanline endpoint of the median sensor is chosen to be the reference scanline endpoint.

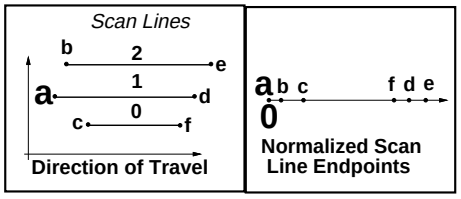

Figure 4: The scanline endpoints are *normalized* by subtracting the position of the reference scanline endpoint (in this case, scanline endpoint "a").

2.4 Configuration

Without loss of generality, we assume the scanning sensor moves along the x axis, making the absolute x position irrelevant. The object's normalized configuration depends only upon y and θ. Configuration (y, θ) corresponds to rotating the object counterclockwise by θ and then translating the object in the y direction by y (refer Figure 5). Indexing entries are synthesized by predicting the discretized normalized intersections of the transformed object with horizontal scanlines (refer Figure 6). To accommodate incorrect models and numerical imprecision, we include additional indexing table entries corresponding to ϵ tolerances.

3 Complete Table Construction

In this section, we detail construction of *complete* indexing tables. The key idea is that each indexing table entry corresponds to a finite number of continuous regions in configuration space; furthermore, these regions are separated by only two types of boundaries: discretization boundaries separating configurations which are discretized to different indexing coordinates (refer Figures 7,10), and correspon-

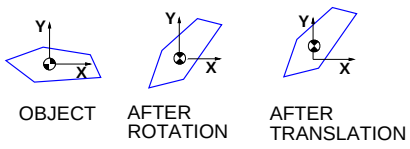

Figure 5: Configuration (θ, y) of object O corresponds to a copy of O rotated about the origin by θ and then translated along the y-axis by y.

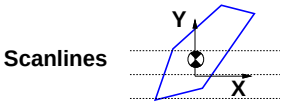

Figure 6: Scanning sensor data is synthesized by predicting the discretized normalized intersections of the transformed object with scanlines.

dence boundaries separating configurations with different correspondences (refer Figures 8,14). Boundaries are curves in this (y, θ) two-dimensional configuration space. We can enumerate all of the indexing table entries by enumerating all of these regions in configuration space; one way to enumerate all of the regions is to enumerate all of the pairwise intersections of the boundary curves.

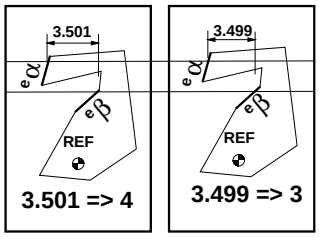

Figure 7: Two different configurations with different indexing coordinates; the normalized scanline endpoints are discretized to different indexing coordinates

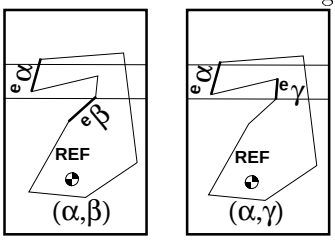

Figure 8: Two different configurations with different feature correspondences

3.1 Algorithm Outline

Definition 1 *Discretization* refers to the process of rounding an indexing coordinate to the closest multiple of the discretization $\delta : \overline{x_i} \rightarrow \lfloor \frac{\overline{x_i} + \frac{\delta}{2}}{\delta} \rfloor$

Definition 2 *Discretization Boundary Curves* characterize configurations for which a predicted normalized scanline endpoint bounds two discretization regions: $\overline{x_i} = k\delta + \frac{\delta}{2} | k \in \mathbb{Z}$.

Definition 3 *Correspondence Boundary Curves* characterize configurations for which a predicted normalized scanline endpoint bounds two correspondence regions; *i.e.*,, the scanline contacts two features at the incident vertex.

1. Enumerate all discretization boundary curves and correspondence boundary curves
2. Enumerate all intersections between these curves
3. Synthesize indexing table entries at intersections

Figure 9 depicts a set of discretization boundary curves and correspondence boundary curves in configuration space. Each region (cell in the arrangement) corresponds to a single indexing table entry. A complete indexing table is produced by synthesizing an indexing table entry for each configuration where two curves intersect; in order to compute intersections algebraically, we use a change of variables from θ to t ($t = \tan(\frac{\theta}{2})$) to transform trigonometric expressions into rational algebraic expressions.

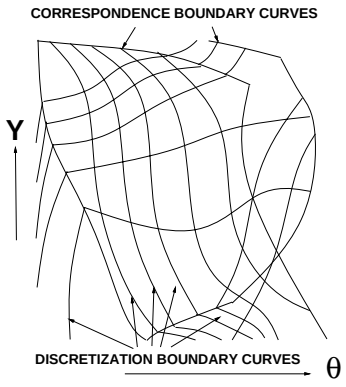

Figure 9: A set of discretization boundary curves and correspondence boundary curves in (θ, y) configuration space.

3.1.1 Discretization Boundary Curves

Discretization boundary curves ($DBC(\theta, y)$, DB-curves) describe configurations in which one of the normalized scanline endpoints is exactly $\overline{x_\alpha} = k\delta + \frac{\delta}{2}$, where $k \in \mathbb{Z}$, *i.e.*, $x_\alpha - x_{Ref} = k\delta + \frac{\delta}{2}$. For example,

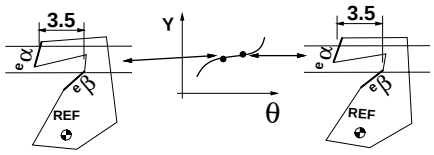

Figure 10: A DB-curve corresponding to edges e_α, e_β with discretization $x_\alpha - x_{Ref} = 3.5$

Figure 10 depicts a DB-curve corresponding to edges e_α, e_β with discretization $x_\alpha - x_{Ref} = 3.5$.

DB-curves are formulated via two intermediate functions. First we formulate the x coordinate of the extended intersection between a horizontal scanline and an arbitrary edge e_α as a function $I(\theta, y)$ (refer Figure 11 and equation (1)). Second, we formulate the difference between two extended intersections $\Delta I(\theta, y)$; and third, we define DB-curves as the set of all configurations for which the difference function is equal to the desired value $DB_{\beta,\alpha}(\theta, y)$.

In Figure 11, e_α is parameterized by α and R_α, where α refers to e_α's outward pointing normal direction, and R_α refers to the minimum distance from the origin to e_α. The horizontal scanline is characterized by its height D_α.

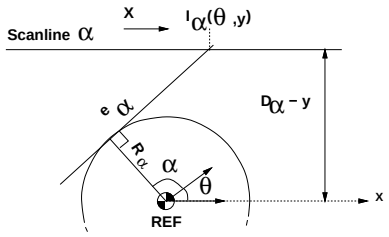

Figure 11: Extended intersection function $I(\theta, y)$ characterizes the x coordinate of the extended intersection between transformed edge segment e_α and scanline α.

$$
\begin{aligned}
I_\alpha(\theta, y) &= \frac{R_\alpha + (y - D_\alpha)\sin(\theta + \alpha)}{\cos(\theta + \alpha)} \qquad (1)\\
&= \frac{R_\alpha + (y - D_\alpha)(\sin\theta\cos\alpha + \cos\theta\sin\alpha)}{\cos\theta\cos\alpha - \sin\theta\sin\alpha}\\
I_\alpha(t, y) &= \frac{R_\alpha(1 + t^2) + (y - D_\alpha)(2t\cos\alpha + (1 - t^2)\sin\alpha)}{(1 - t^2)\cos\alpha - 2t\sin\alpha}\\
&= \frac{y(-t^2\sin\alpha + 2t\cos\alpha + \sin\alpha)}{(1 - t^2)\cos\alpha - 2t\sin\alpha} +\\
&\quad \frac{t^2(R_\alpha - D_\alpha\sin\alpha) - 2tD_\alpha\cos\alpha + R_\alpha + D_\alpha\sin\alpha}{(1 - t^2)\cos\alpha - 2t\sin\alpha}
\end{aligned}
$$

Extended intersection difference functions (ΔI) characterize the difference between extended intersection functions (refer Figure 12 and equation (2)). Discretization boundary curves are the set of configurations where the extended intersection difference function is equal to the desired value, or, in other words, the zero set of the difference between the extended intersection difference function and the desired value, $\Delta_{\beta,\alpha} = -k\delta + \frac{\delta}{2}$. Cross multiplying by the denominators of the two extended intersection difference functions yields an algebraic expression, $DBC(\theta, y)$ (refer equations (4,5)).

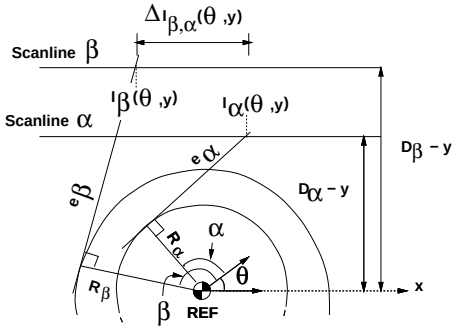

Figure 12: Extended intersection difference functions $\Delta I_{\beta,\alpha}(\theta, y)$

$$
\begin{aligned}
\Delta I_{\beta,\alpha}(\theta, y) &= I_\beta(\theta, y) - I_\alpha(\theta, y) \qquad (2)\\
&= \frac{R_\beta + (y - D_\beta)(\sin\theta\cos\beta + \cos\theta\sin\beta)}{\cos\theta\cos\beta - \sin\theta\sin\beta} -\\
&\quad \left(\frac{R_\alpha + (y - D_\alpha)(\sin\theta\cos\alpha + \cos\theta\sin\alpha)}{\cos\theta\cos\alpha - \sin\theta\sin\alpha}\right)\\
\Delta I_{\beta,\alpha}(t, y) &= \frac{y((1 - t^2)\sin\alpha + 2t\cos\alpha)}{(1 - t^2)\cos\alpha - 2t\sin\alpha} +\\
&\quad \frac{t^2(R_\alpha - D_\alpha\sin\alpha) - 2tD_\alpha\cos\alpha + R_\alpha + D_\alpha\sin\alpha}{(1 - t^2)\cos\alpha - 2t\sin\alpha} -\\
&\quad \left(\frac{y((1 - t^2)\sin\beta + 2t\cos\beta)}{(1 - t^2)\cos\beta - 2t\sin\beta} +\right.\\
&\quad \left.\frac{t^2(R_\beta - D_\beta\sin\beta) - 2tD_\beta\cos\beta + R_\beta + D_\beta\sin\beta}{(1 - t^2)\cos\beta - 2t\sin\beta}\right) \qquad (3)
\end{aligned}
$$

$$
\begin{aligned}
DBC_{\beta,\alpha}(t, y) &= (\Delta I_{\beta,\alpha}(t, y) - \Delta_{\beta,\alpha}) \qquad (4)\\
&((1 - t^2)\cos\beta - 2t\sin\beta)((1 - t^2)\cos\alpha - 2t\sin\alpha)\\
&\quad = y(1 + t^2)^2\sin(\alpha - \beta)\\
&- \Delta_{\beta,\alpha}((1-t^2)\cos\alpha - 2t\sin\alpha)((1-t^2)\cos\beta - 2t\sin\beta) +\\
&((1-t^2)\cos\beta - 2t\sin\beta)((1+t^2)(R_\alpha - D_\alpha\sin\alpha) - 2tD_\alpha\cos\alpha)\\
&+((1-t^2)\cos\beta - 2t\sin\beta)y((1-t^2)\sin\alpha + 2t\cos\alpha) -\\
&((1-t^2)\cos\alpha - 2t\sin\alpha)((1+t^2)(R_\beta - D_\beta\sin\beta) - 2tD_\beta\cos\beta)\\
&-((1-t^2)\cos\alpha - 2t\sin\alpha)y((1-t^2)\sin\beta + 2t\cos\beta) \qquad (5)
\end{aligned}
$$

Although the DBC equations correspond to infinitely long lines, DB-curves correspond to finite length edge segments. The endpoints of the edge segments restrict the orientation range of DB-curves (refer Figure 13), and this orientation range depends solely on the edges and the offset between the discretized contact point and the origin $(\Delta_{\beta,\alpha}, D_\beta - D_\alpha)$.

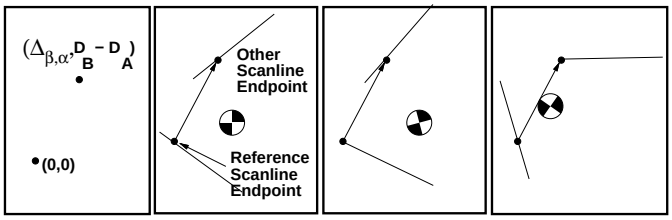

Figure 13: The actual endpoints of the edge segments restrict the orientation range of DB-curves.

3.1.2 Correspondence Boundary Curves

Correspondence boundary curves ($CBC(\theta, y)$, CB-curves) characterize configurations for which the scanline intersects two features at the incident vertex. For example, Figure 14 depicts a CB-curve corresponding to scanline intersecting vertex v_γ, and depend only upon the vertex v_γ, and the scanline height S (refer Figure 15 and equation (6)). Every CB-curve spans the entire orientation range $[0, 2\pi\rangle$ because a rotated vertex can always be translated onto the scanline via translation in y.

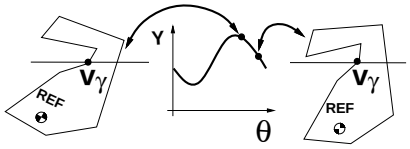

Figure 14: A CB-curve corresponding to the scanline contacting vertex v_γ.

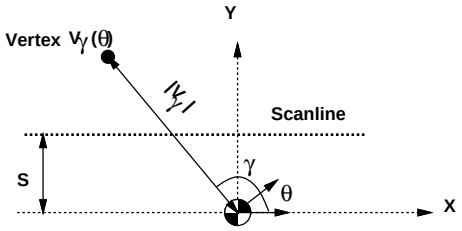

Figure 15: Model used for characterizing correspondence boundary curves

$$
\begin{aligned}
y(\theta) &= |V_\gamma| \sin(\theta + \gamma) - S \qquad (6)\\
CBC_\gamma(\theta, y) &= |V_\gamma| \sin(\theta + \gamma) - S - y \\
CBC_\gamma(t, y) &= |V_\gamma|(2t\cos(\gamma) + (1 - t^2)\sin(\gamma)) \\
&\quad - (1 + t^2)(S + y)
\end{aligned}
$$

3.1.3 Intersecting DB-Curves

This section describes how we compute the intersections of DB-curves (without loss of generality, α, β, γ are the edges and α is the reference edge $DBC_{\beta,\alpha}(t, y) = DBC_{\omega,\alpha}(t, y) = 0$). In the general case, all three of the edges are distinct. The degenerate case is when two of the edges are the same, and the orientation θ can be computed by finding the orientation which places two contact points on the edge.

The general case is handled as follows. Notice that two of three y dependent terms in both $DBC_{\beta,\alpha}(t, y)$ and $DBC_{\gamma,\alpha}(t, y)$ cancel each other out leaving $y \sin(\alpha - \beta)(1 + t^2)^2$ and $y \sin(\alpha - \gamma)(1 + t^2)^2$ as the only y-dependent terms. Cross-multiplying $DBC_{\beta,\alpha}(t, y)$ by $\sin(\alpha - \gamma)$ and $DBC_{\gamma,\alpha}(t, y)$ by $\sin(\alpha - \beta)$ results in both functions having the same y-dependent term; therefore the difference between these two expressions is an expression solely in t (refer equation (7)). Further simplification results in a quadratic expression in t which can be solved exactly, and the y translation is computed given t.

$$
\begin{aligned}
&\underbrace{\sin(\alpha - \beta) y (1 + t^2)^2 + DBC^{*y}_{\beta,\alpha}(t)}_{DBC_{\beta,\alpha}(t,y)} = 0 = \\
&\underbrace{\sin(\alpha - \gamma) y (1 + t^2)^2 + DBC^{*y}_{\gamma,\alpha}(t)}_{DBC_{\gamma,\alpha}(t,y)} \\
&\sin(\alpha - \gamma)(\sin(\alpha - \beta) y (1 + t^2)^2 + DBC^{*y}_{\beta,\alpha}(t)) = 0 \\
&= \sin(\alpha - \beta)(\sin(\alpha - \gamma) y (1 + t^2)^2 + DBC^{*y}_{\gamma,\alpha}(t)) \\
&\Rightarrow -\sin(\alpha - \beta) DBC^{*y}_{\beta,\alpha}(t) = -\sin(\alpha - \gamma) DBC^{*y}_{\gamma,\alpha}(t) \\
&\Rightarrow a_2 t^2 + a_1 t + a_0 = 0
\end{aligned}
$$

3.1.4 Intersecting CB-Curves and DB-Curves

The orientation of the intersection of DB-curves and CB-curves is found by substituting equation (6) into equation (3), and cross multiplying producing an equation solely in t which can be solved numerically. The y translation is computed given t.

3.1.5 Intersecting CB-Curves

The orientation θ of the intersection of two CB-curves is determined by finding the two orientations for which the vector between the vertices has the desired y component. The y translation is computed given θ.

3.2 Complete Lookup Table Sizes

Complete indexing tables were constructed for multiple objects at various discretizations. For various objects shown in Figure 17, Table 1 lists the sizes of indexing tables, the number of model features n, and κ, the average number of interpretations for each valid indexing coordinate. The tables were constructed for a three beam sensor. Since recognition time was a function of κ, all of the non-rectangular objects took longer than five microseconds to identify, but, in every case, increased speed can be bought by constructing tables with finer discretization.

Object	Disc.	n	Table Size	κ
Rectangle	2.5	4	13059	1.694
Rectangle	1.0	4	59856	1.232
Camera Part	2.5	6	39579	2.112
Camera Part	1.0	6	161104	1.490
Braket	2.5	16	87297	1.553
Braket	1.5	16	218986	1.296
Key	2.5	19	35416	3.054
Key	1.5	19	119508	1.769
Credit Card	2.5	4	26611	1.503
Credit Card	1.0	4	140427	1.237
Octagonal Braket	2.5	8	61106	2.498
Octagonal Braket	1.5	8	135878	1.920

Table 1: Sizes of complete indexing tables for various objects, resolutions, and number of scanning beams.

4 Localization Technique

In this section we outline an algorithm for estimating an object's pose given the sensed data and the correspondence interpretation. Most of the previous work in pose estimation deals with matching features of similar type, such as points to points, or lines to lines. For scanline data, scanline endpoints correspond to non-point features. The optimal pose estimate is computed using the localization algorithm described by Wallack and Manocha [12] which finds the rigid two-dimensional transformation which optimally transforms the non-point model features onto data points (refer Figure 16).

Their technique computes the transformation which minimizes the sum squared errors between the transformed points and the non-point features, where the error corresponds to the minimum distance between a point and a feature. The global minimum is found algebraically by formulating an algebraic error expression for the sum squared errors between the point set and the set of linear features parameterized by (x, y, θ). Local methods for computing the global minimum, such as Gauss-Newton's method, are susceptible to problems of local minima, and these problems are exacerbated for small data sets. Wallack and Manocha's technique computes the global minimum by finding all the local extrema of the error function and then checking each of these extrema, since the global minimum is a member of the local extrema. All of the local extrema are found by solving the system of partial derivative equations. They use a combination of algebraic and numerical techniques to solve the system of partial derivative equations one variable at a time via resultants and matrix computations. This technique is general and is applicable to circular features as well.

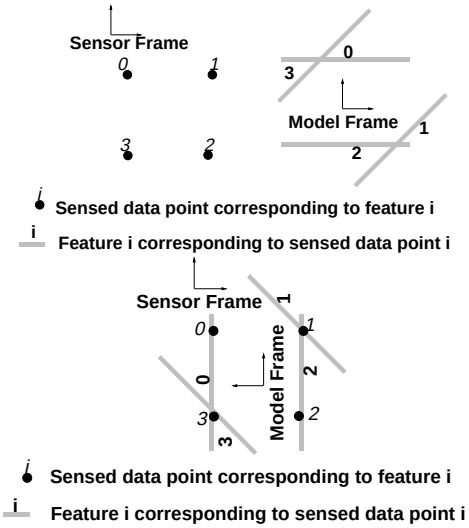

Figure 16: Given a set of sensed data points in the sensor reference frame, and non-point model features in a model reference frame (top), the task is to determine the model position, which is the transformation which makes the features converge (bottom)

5 Experiment and Results

In this section, we discuss the results of experiments which analyzed the technique's recognition and localization performance. Positional and orientational repeatabilities were measured by localizing a precisely modeled rigidly-held object (a white rectangle imprinted on a piece of paper) multiple times along various scanning paths. The rectangle was printed using a laser printer and provided a precise, accurately modeled object for measuring the technique's performance. Recognition performance was measured by identifying known objects (Figure 17) and a similarly sized rectangles in a variety of poses.

Figure 17: The set of possible objects.

5.1 Positional Accuracy Results

The object was localized many times in order to determine the technique's localization repeatability. The object traversed many parallel paths, and traversed each path 25 times.The average localized positions, and standard deviations for all 25 localized positions are given in Table 2. The average localized position was computed for each path, and these averages agreed to within ± 0.01 mm. The standard deviations σ for each path were approximately 0.01 mm; a 2σ error bound assumption implies that localized position should vary by 0.02 mm.

path x-position (mm)	mean localized part position (mm)	std. dev. σ (mm)
400.00	(2.407, 3.270)	0.0061
410.00	(2.398, 3.276)	0.0067
420.00	(2.413, 3.278)	0.0077
430.00	(2.418, 3.276)	0.0077
440.00	(2.390, 3.286)	0.0111

Table 2: The mean localized positions, and the standard deviations of those (25) localized positions resulting from localizing a rectangle along different x ordinate paths.

5.2 Orientational Accuracy Results

The localization technique's orientational accuracy was measured by localizing the rectangle at known relative orientations. The differences between the end-effector's commanded orientation and the rectangle's computed orientation are given in Table 3. The difference between the commanded orientations and the average localized orientations differed by less than $\pm 0.03^o$.

mean robot orientation θ ($^\circ$)	mean localized ($^\circ$)	std. dev. σ ($^\circ$)
72.0089	90.1853	0.0171
36.0076	90.2400	0.0224
-0.1017	90.1840	0.0233
-36.0649	90.2039	0.0266
-71.9008	90.2198	0.0270

Table 3: Results: The commanded orientations, the mean estimated orientations, and the standard deviations of those (25) estimated orientations positions resulting from localizing a rectangle at different orientations.

5.3 Recognition Results

The objects shown in Figure 17 and four similarly sized rectangles $(75 \times 19, 100 \times 20, 125 \times 21, 150 \times 22)$ were used to measure the scanning sensor's recognition performance by identifying each object 50 times in a variety of poses. The technique correctly identified each object every time.

6 Conclusion

In this paper, we presented a model-based object recognition and localization technique based on scanning beam sensors. Beam sensors are highly accurate, robust, fast, and inexpensive due to their simple specification. In order to utilize indexing to solve the correspondence problem in constant time, we constructed complete indexing tables. Objects were localized using a combination of algebraic and numerical methods to match the scanline endpoints to the non-point model features; positional estimates were repeatable to 0.025 mm, and orientational estimates were repeatable to 0.1^o.

Acknowledgements

The authors wish to acknowledge: Dr. Dinesh Manocha for valuable help in producing this paper, and Francesca Barrientos for helpful criticisms and suggestions on this report.

References

[1] John Canny and Ken Goldberg. RISC robotics. In *IEEE International Conference on Robotics and Automation*, 1994.

[2] D. T. Clemens and D. W. Jacobs. Space and time bounds on indexing 3-d models from 2-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1007–1017, 1991.

[3] O. D. Faugeras. *Three Dimensional Computer Vision - A Geometric Perspective*. MIT Press, 1993.

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

[5] D. Forsyth, L. Mundy, A. Zisserman, A. Heller, and C. Rothwell. Invariant descriptors fo 3-d object recognition and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:971–991, Oct 1991.

[6] W. Eric L. Grimson. Sensing strategies for disambiguating among multiple objects in known poses. *International Journal of Robotics Research*, RA-2(4):196–213, 1986.

[7] Daniel P. Huttenlocher and Shimon Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.

[8] Alan Kalvin, Edith Schonberg, Jacob T. Schwartz, and Micha Sharir. Two-dimensional model-based boundary matching using footprints. *International Journal of Robotics Research*, 5(4):38–55, 1986.

[9] Yehezkel Lamdan, Jacob T. Schwartz, and Haim J. Wolfson. Object recognition by affine invariant matching. *International Conference of Computer Vision*, 1988.

[10] Jacob T. Schwartz and Micha Sharir. Identification of partially obscured objects in three dimensions by matching noisy characteristic curves. *International Journal of Robotics Research*, 6(2):29–44, 1987.

[11] Aaron Wallack and John Canny. An indexing technique for sparse sensing strategies. submitted to the 1995 IEEE International Symposium on Assembly on Task Planning, 1995.

[12] Aaron Wallack and Dinesh Manocha. Object localization using algebraic techniques. (preprint University of California, Berkeley), 1994.