[19] D. W. Thompson and J. L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. *IEEE International Conference on Robotics and Automation*, 1:208–220, 1987.

[20] Aaron Wallack, John Canny, and Dinesh Manocha. Object recognition and localization using optical sensors. (preprint, University of California, Berkeley), 1992.

[21] Jean Ponce, Anthony Hoogs, and David J. Kriegman. On using cad models to compute the pose of curved 3d objects. *CVGIP: Image Understanding*, 55(2):184–197, 1992.

[22] Aaron Wallack and John Canny. Object localization using light beam sensing. Technical Report 92-14, Engineering Systems Research Center, 1992.

[23] Balas K. Natarajan. Some paradigms for the automated design of parts feeders. *International Journal of Robotics Research*, 8(6):98–109, December 1986.

| $\mathcal{R}_1$ | | $\mathcal{R}_4$ | |
| --- | --- | --- | --- |
| edge-edge pair: <orientation range> $\mathcal{R}_1$ | orientation range - $\angle\mathcal{R}_1$ | edge-edge pair: <orientation range> $\mathcal{R}_4$ | orientation range - $\angle\mathcal{R}_4$ |
| $\{[5,1]:<0.129,\ -0.129>\}$ | $<0.046,\ -0.212>$ | $\{[5,4]:<-1.571,\ -2.261>\}$ | $<0.544,\ -0.146>$ |
| $\{[5,0]:<0.132,\ -0.125>\}$ | $<0.049,\ -0.208>$ | $\{[5,4]:<-1.571,\ -2.261>\}$ | $<0.544,\ -0.146>$ |
| $\{[0,3]:<-0.255,\ -0.206>\}$ | $<-0.338,\ -0.349>$ | $\{[0,5]:<-2.261,\ 3.142>\}$ | $<-0.146,\ -1.027>$ |
| $\{[0,2]:<-0.131,\ -0.206>\}$ | $<-0.214,\ -0.349>$ | $\{[0,5]:<-2.261,\ 3.142>\}$ | $<-0.146,\ -1.027>$ |
| $\{[0,1]:<-0.125,\ -0.132>\}$ | $<0.049,\ 0.042>$ | $\{[0,5]:<-2.261,\ 3.142>\}$ | $<-0.146,\ -1.027>$ |
| $\{[3,5]:<3.011,\ 2.876>\}$ | $<2.928,\ 2.793>$ | $\{[3,2]:<1.571,\ 0.0>\}$ | $<3.686,\ 2.115>$ |
| $\{[3,4]:<3.142,\ 3.010>\}$ | $<3.059,\ 2.71>$ | $\{[3,2]:<1.571,\ 0.0>\}$ | $<3.686,\ 2.115>$ |
| $\{[3,0]:<2.877,\ 2.876>\}$ | $<2.794,\ 2.793>$ | $\{[3,2]:<1.571,\ 0.0>\}$ | $<3.686,\ 2.115>$ |

Table 5: The consistent interpretations of $\mathcal{R}_1$'s edge-edge pairs and $\mathcal{R}_4$'s edge-edge pairs.

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

[5] 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.

[6] Yehezkel Lamdan and Haim J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. Technical Report No. 368, New York University, Robotics Research Laboratory, Department of Computer Science, May 1988.

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

[8] W. Eric L. Grimson. The combinatorics of heuristic search termination for object recognition in cluttered environments. *IEEE Transactions on Pattern Matching and Machine Intelligence*, 13(9):920-935, 1991.

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

[10] Anil S. Rao and Kenneth Y. Goldberg. Orienting generalized polygonal parts. In *IEEE International Conference on Robotics and Automation*, pages 2263-2268, 1992.

[11] Kenneth Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 1992.

[12] Aaron Wallack, John Canny, and Dinesh Manocha. Object localization using crossbeam sensing. In *IEEE International Conference on Robotics and Automation*, 1993.

[13] Dinesh Manocha, Aaron Wallack, and John Canny. Object recognition and localization using optical sensors. (preprint, University of California, Berkeley), 1992.

[14] D. Manocha. *Algebraic and Numeric Techniques for Modeling and Robotics*. PhD thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1992.

[15] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[16] 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.

[17] Bernard Klaus Paul Horn. *Robot Vision*. McGraw-Hill, seventh edition, 1989.

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

| $\delta$ | direction | edge-edge pair < orientation range > | |
|---|---|---|---|
| 47.9 | counterclockwise | { [0, 3] :< −0.266, −0.266 >}<br>{ [0, 1] :< −0.125, −0.132 >}<br>{ [5, 0] :< 0.132, 0.125 >}<br>{ [3, 0] :< 0.266, 0.266 >} | { [0, 2] :< −0.132, −0.266 >}<br>{ [5, 1] :< 0.125, 0.125 >}<br>{ [4, 0] :< 0.266, 0.132 >} |
| 47.9 | clockwise | { [0, 3] :< −0.266, −0.266 >}<br>{ [4, 3] :< 0.0, −0.132 >}<br>{ [3, 2] :< 0.132, 0.0 >}<br>{ [3, 0] :< 0.266, 0.266 >} | { [5, 3] :< −0.132, −0.266 >}<br>{ [3, 3] :< 0.0, 0.0 >}<br>{ [3, 1] :< 0.266, 0.132 >} |
| 48.1 | counterclockwise | { [0, 3] :< −0.265, −0.265 >}<br>{ [0, 1] :< −0.129, −0.131 >}<br>{ [5, 0] :< 0.131, 0.129 >}<br>{ [3, 0] :< 0.265, 0.265 >} | { [0, 2] :< −0.131, −0.265 >}<br>{ [5, 1] :< 0.129, −0.129 >}<br>{ [4, 0] :< 0.265, 0.131 >} |
| 47.9 | clockwise | { [0, 3] :< −0.265, −0.265 >}<br>{ [4, 3] :< 0.0, −0.131 >}<br>{ [3, 2] :< 0.131, 0.0 >}<br>{ [3, 0] :< 0.265, 0.265 >} | { [5, 3] :< −0.131, −0.265 >}<br>{ [3, 3] :< 0.0, 0.0 >}<br>{ [3, 1] :< 0.265, 0.131 >} |

Table 2: Edge-edge pairs resulting from walking in different directions for different $\delta$ length chords.

| edge-edge pair < orientation range > | | |
|---|---|---|
| { [0, 3] :< −0.265, −0.266 >} | { [0, 2] :< −0.131, −0.266 >} | { [0, 1] :< −0.125, −0.132 >} |
| { [5, 1] :< 0.129, −0.129 >} | { [5, 0] :< 0.132, 0.125 >} | { [4, 0] :< 0.266, 0.131 >} |
| { [5, 3] :< −0.131, −0.266 >} | { [4, 3] :< 0.0, −0.132 >} | { [3, 3] :< 0.0, 0.0 >} |
| { [3, 2] :< 0.132, 0.0 >} | { [3, 1] :< 0.266, 0.132 >} | { [3, 0] :< 0.266, 0.265 >} |

Table 3: The edge-edge pairs from $\mathcal{PPE}([47.9, 48.1])$ in the continuous family of chords containing $\theta = 0$.

$$(1 + t^2) \begin{vmatrix} a & b \\ d & e \end{vmatrix} s(t) =$$

$$(-acd + a\cos\theta_0 d\delta + b\cos\theta_0 d\delta - bce + af + b^2f \, 2\delta e\sin\theta_0 + b^2\delta e\sin\theta_0) +$$

$$t(2a^2\cos\theta_0\delta e + 2b^2\cos\theta_0\delta e - 2ad\delta\sin\theta_0 - 2b^2d\delta\sin\theta_0) +$$

$$t^2(-acd - a^2\cos\theta_0 d\delta - 2b\cos\theta_0 d\delta - bce^2 + af + b^2f - a^2\delta e\sin\theta_0 - b^2\delta e\sin\theta_0) \tag{11}$$

# 6   REFERENCES

[1] John Canny and Ken Goldberg, RISC robotics. (preprint, University of California, Berkeley).

[2] Richard Cole and Chee K. Yap. Shape from probing. *Algorithmica*, 8:19-38, March 1987.

[3] Nicholas Ayache and Oliver D. Faugeras.  Hyper:  A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Matching and Machine Intelligence*, 8(1):44-54, 1986.

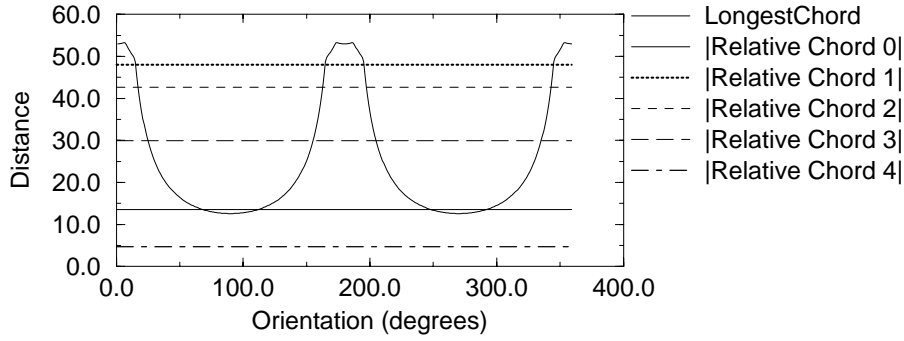| edge-edge pair < orientation range > | | |
|---|---|---|
| { [0, 1] :< 0.0, −0.881 >} | { [4, 3] :< 0.0, −1.571 >} | { [1, 2] :< −0.881, −1.571 >} |
| { [5, 4] :< −1.571, −2.261 >} | { [2, 3] :< −1.571, 3.142 >} | { [0, 5] :< −2.261, 3.142 >} |
| { [3, 4] :< 3.142, 1.571 >} | { [1, 0] :< 3.142, 2.261 >} | { [2, 1] :< 2.261, 1.571 >} |
| { [4, 5] :< 1.571, 0.881 >} | { [3, 2] :< 1.571, 0.0 >} | { [5, 0] :< 0.881, 0.0 >} |

Table 4: $\mathcal{PPE}([4.58, 4.78])$

Figure 21: The adapter's $\mathcal{LC}$ function superimposed with the lengths of the relative chords.
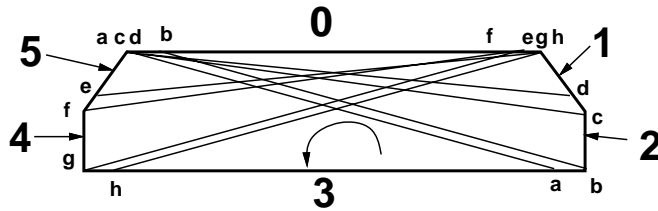


Figure 22: Enumerating all of the chords in the continuous family of chords $above\ the\ Longest\ Chord$.

described in section 3.3.2

## 5  CONCLUSION

In this paper we described a correspondence algorithm for scan line data given polyhedral objects. The algorithm was specialized for convex polyhedral objects. This research is a successful example of the RISC Robotics paradigm which attempts to solve manufacturing problems with suitable hardware: light beam scanning uses accurate, robust, cost-effective sensors. The algorithm generates candidate pairs of edges from individual scan line data and tests for consistent combinations by brute force or intersecting parameterized ranges. The hypotheses are orientationally ordered (by $\theta$), and therefore testing combinations only takes $O(n+A)$ time where $n$ is the object's complexity and $A$ is the number of chord matches.

**Acknowledgements**

**Appendix**

**Intersection Point Function** $s_i(t)$

In Figure 18, let $a$, $b$, $c$ define the line containing edge $A$ ($ax + by = c$), and $d$, $e$, $f$ define the line containing edge $B$ ($dx + ey = f$) translated by a vector of length $\delta$ and orientation $\theta + \theta_0$ ($t = \tan(\frac{theta}{2})$). The translated edge $B'$ is defined by $d$, $e$, $f'$ where $f'$ is given in equation 9. The intersection $s(t)$ is given in equation (10), and the expanded algebraic equation for $s(t)$ is given in equation (11).

$$f' = f + \frac{1}{1+t^2}\delta\left(d\left[(1-t^2)\cos\theta_0 - 2t\sin\theta_0\right] + e\left[2t\cos\theta_0 + (1+t^2)\sin\theta_0\right]\right) \quad (9)$$

$$s(t) = \frac{-b\begin{vmatrix} c & b \\ f' & e \end{vmatrix} + a\begin{vmatrix} a & c \\ d & f' \end{vmatrix}}{\begin{vmatrix} a & b \\ d & e \end{vmatrix}} \quad (10)$$
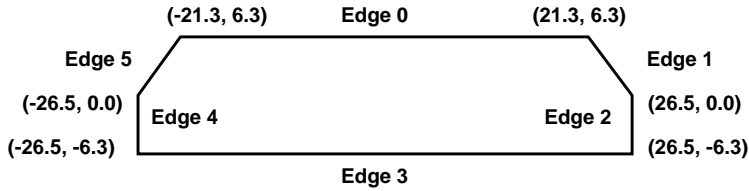
**(-21.3, 6.3)**    **Edge 0**    **(21.3, 6.3)**

**Edge 5**    **Edge 1**

**(-26.5, 0.0)**    **(26.5, 0.0)**

**Edge 4**    **Edge 2**

**(-26.5, -6.3)**    **(26.5, -6.3)**

**Edge 3**

Figure 19: The shadow of an adapter which is used in this example.

**(4,-8.39)**    **(4,26.51)**

**(0,-21.32)**    $R_0$    $R_1$    $R_2$

**(-4,-23.74)**    $R_4$    $R_3$    **(0,21.32)**
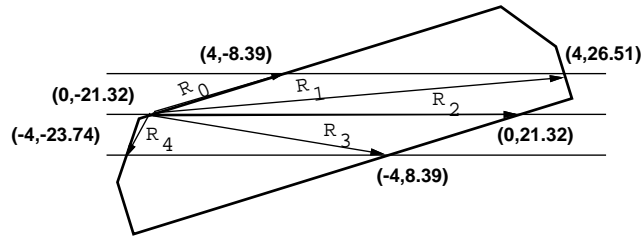
**(-4,8.39)**

Figure 20: The scanline endpoints of the adapter rotated by 17. 88$^\circ$.

## 4.1 Computing $\mathcal{PPE}$ for $\mathcal{R}_1$

For $\mathcal{R}_1$ ($|\mathcal{R}_1| = 48$), $\mathcal{PPE}$ ($[|\mathcal{R}| \pm \epsilon]$) contains 23 distinct edge-edge-pairs. For this reason, we only sketch, rather than trace, the construction technique. Computing $\mathcal{PPE}$ ($[|\mathcal{R}_1| \pm \epsilon]$) involves computing $\mathcal{PPE}$ ($\Delta_{min} = 47.9$) and computing $\mathcal{PPE}$ ($\Delta_{max} = 48.1$).

From Figure 21, we see that there are two continous families of chords of length 47.9. The first step in computing $\mathcal{PPE}$ (47.9) is scanning down $\mathcal{LC}(\theta)$ and finding an < ExtremalVertex, ExtremalEdge> pair which contacts a chord larger than 47.9. There are four orientations $\theta_{extremal}$ for which $\mathcal{LC}(\theta_{extremal}) = 47.9$. These are: $\{\pm 0.2618, \pm 2.8741\}$ radians. They all contact the parallel edges $edge_0$ and $edge_3$ (refer Figure 19).

In this paragraph, we sketch how the edge-edge pairs in continous family of chords containing the orientation $\theta = 0$ are collected. Initially, we begin with $c_{initial}$ of orientation $-0.2618$ radians. From this chord, we walk in two different directions (clockwise and counterclockwise) around the boundary in order to enumerate the pairs of edges "above" and "below" the Longest Chord. Figure 22 shows the trail of counterclockwise rotating chord as it enumerates all of the edge-edge pairs in the continous subfamily of chords. The counterclockwise traversal enumerated the following edge-edge pairs and orientation ranges (denoted by [bottomedge, top edge]: < angle range>) are given in Table 2, as well as the edge-edge pairs from other chord traversals.

The possible pairs of edges, in that continous family, consistent with the sensed chord ($\mathcal{PPE}$ ($[|\mathcal{R}_1| \pm \epsilon]$) are computed by merging the edge-edge pairs from Table 2. $\mathcal{PPE}$ is given in Table 3, and $\mathcal{PPE}$ ($[|\mathcal{R}_4| \pm \epsilon]$) is given in Table 4

At this point, we would subtract the angle $\angle \mathcal{R}_i$ from the orientation ranges for each edge-edge pair. Table 5 lists the possible interpretations for which the 'bottom'edges are consistent. The two chords $\mathcal{R}_1, \mathcal{R}_4$ constrain the set possible interpretations. As more constraints are included, all of the inconsistent interpretations disappear, leaving only the *real* interpretation and a nearly interpretation (for which the scanline origin resides on edge 5). The two interpretations can be compared by computing the optimal pose for each, or the heuristic approach

| scanline data point | $\mathcal{R}_i$ | $|\mathcal{R}|$ | $\angle \mathcal{R}_i$ (radians) |
|---|---|---|---|
| (4,-8.39) | (4, 12.93) | 13.53 | 0.3 |
| (4,26.51) | (4, 47.83) | 48.00 | 0.083 |
| (0,21.32) | (0, 42.64) | 42.64 | 0.0 |
| (-4,8.39) | (-4, 29.71) | 29.98 | -0.134 |
| (-4,-23.74) | (-4, -2.42) | 4.68 | -2.115 |

Table 1: The relative chords corresponding to each of the scanline data points.

$$Error(t) = \min_t \sum_i \sum_{j<i} (s_i(t) - s_j(t))^2 = \frac{Q(t)}{(1+t^2)^2} \tag{7}$$

$$\frac{dError(t)}{dt} = \frac{1}{(1+t^2)^3}((1+t^2)\frac{dQ(t)}{dt} - 2t\,Q(t)) \tag{8}$$
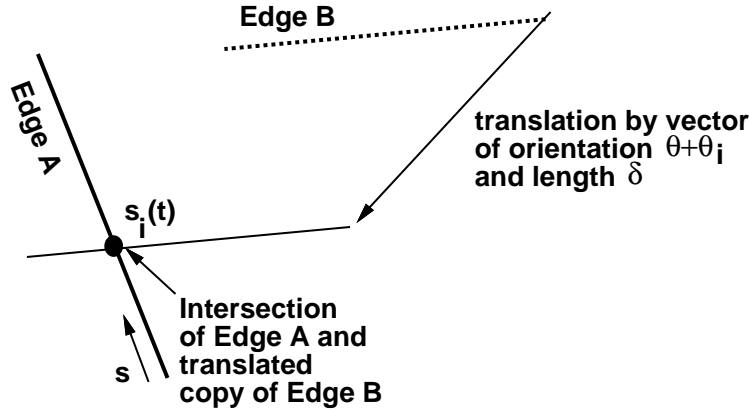


Figure 18: The chord's contact points with the two contact edges can be found by translating one of the edges by the chord, and finding the intersection.

## 3.4 Algorithmic complexity

For each orientation $\theta$ there are $2^{2k-1}$ sets of edges assuming that $|\mathcal{PPE}_i| = 2$ (the normal case) and $k$ is the number of scanning beams. $|\mathcal{PPE}|$ is usually $O(n)$ since $\mathcal{PPE}$ normally consists of a small number chord family, and each family includes $O(n)$ pairs of edges.

orientations with different $\mathcal{PPE}$ sets, where $n$ is the complexity of the object.

### 3.4.1 Analysis

For convex objects, the sets $\mathcal{PPE}$ are ordered by $\theta$, and there are at most 2 pairs of edges for each of the $2k - 1$ relative chords. In the worst case, $A$, the number of feasible combinations could be as much as $2^{2k-1}$. Since we are given high-precision data, we can assume that $2k - 1$ values are overkill; intersecting a fixed number $2k - 1 \geq l \geq 2$ of relative chords suffices to specify the pose. This approach only needs to compare $2^l |\mathcal{PPE}|$ pairs of edges. $|\mathcal{PPE}|$ is usually $O(n)$, so the normal running time for the complete algorithm is $O(2^{2k-1} n)$, and $O(2^l n)$ for the *optimistic* approach.

For non-convex objects, constructing $\mathcal{PPE}$ requires $O(n^2)$ time, and sorting these sets by $\theta$ takes $O(n^2 \log n)$ time. Merging these sets takes $O(A + \sum |\mathcal{PPE}|)$ time, where $A$ is the number of feasible matches. Unlike convex objects, for which each orientation corresponds to at most two chords, we give no upper bound on the number of chords corresponding to each orientation for non-convex objects. In the worst case, $A$ the number of feasible match combinations produced by merging two $\mathcal{PPE}$ sets from non-convex shadows, can be as large as $O(n^3)$ (since each scanline origin can only generates $O(n)$ feasible edge-edge pairs).

## 4 EXAMPLE

In order to help describe the correspondence algorithm, we run through an example in this section. The scanlines produced from scanning the adapter shown in Figure 19 are shown in Figure 20.

The first step is to choose a scanline origin. The middle-left scanline point $(0, -21.32)$ is chosen as the *scanline origin*; the relative chords are given in Table 1. Since the object is convex, we execute the *efficient* $\mathcal{PPE}$ algorithm (as oppose to brute force enumeration). The next step is to compute $\mathcal{PPE}$ for each relative chord. Computing $\mathcal{PPE}$ $(||R_i \pm \epsilon|])$ involves finding the extreme chords of $\delta$ length continous chord families, and then walking through each family of chords. Figure 21 shows the lengths of the relative chords overlayed with the object's *Longest Chord* function. We conservatively assume that the total error is less than $0.1$ $\epsilon = 0.1$.

### 3.3.1 Determining the object's pose from two relative chords ($\mathcal{R}_i$, $\mathcal{R}_j$)

We can compute geometrically the orientation $\theta$ for which two relative chords contact three edges ($\mathcal{R}_i$, $\mathcal{R}_j$). Consider a triangle $T$ whose the three vertices correspond to the scanline origin and the two relative chords. The intersection of the two curves corresponds to the intersection of those three points onto their respective edges. That corresponds to fitting a triangle $T$ into a wedge formed by the three corresponding edges. Solving equations 5 and 6 for the free variables $a$ and $c$ determines the orientation $\theta$.
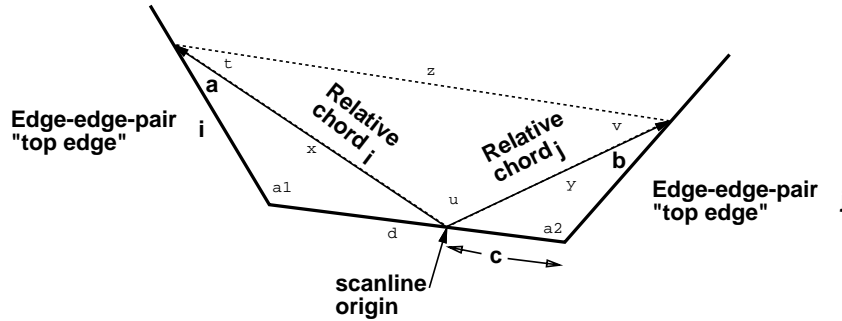


Figure 17: Solving for the intersection of three scanlines can also be viewed geometrically.

$$\frac{\sin a}{d - c} = \frac{\sin a_1}{x}$$

$$\frac{\sin b}{c} = \frac{\sin a_2}{y}$$

$$a + b = \pi - a_1 - a_2 - t - v \quad \rightarrow \quad b = \pi - a_1 - a_2 - t - v - a$$

$$x \sin a = (d - c)\sin a_1 \tag{5}$$

$$y \sin(K - a) = c \sin a_2 \tag{6}$$

### 3.3.2 Consistency heuristic for edge-edge pair interpretations

In this section, we describe a relatively fast heuristic for validating a set of scanline interpretations by computing an optimal orientation $\theta$. This heuristic does not compute the optimal pose, and therefore, does not provide mathematically rigorous information about the quality of the correspondence/interpretation. But we rely on this *non-rigorous* heuristic because the *pseudo-optimal* $\theta$ is estimated more quickly than *optimal* $\theta$. This should suffice since the data are very precise, and since the scanline data should closely match the model.

The heuristic is defined as the sum squared distances between the scanline origins for each relative chord and edge-edge pair. As a prerequisite, all of the edge-edge pairs should consistently have the same "bottom edge" corresponding to the scanline origin. Note that finding a chord's endpoints on edges $e_i$, $e_j$ can be found by intersecting $e_i$ with a copy of $e_j$ translated by the chord.

We define algebraic functions of the positions of the scanline origin for each relative-chord/edge-edge-pair; that scanline origin corresponds to the intersection point (refer Figure 18) between the "bottom edge" and the "top edge" translated by a vector of length $|\mathcal{R}|$ and orientation $\theta + \angle\mathcal{R}$. In order to construct an algebraic function, we utilize the trigonometric substitution $t = \tan(\frac{\theta}{2})$.

The intersection point can be defined by a rational function $s_i(t) = \frac{S(t)}{1 + t^2}$, where the numerator is a quadratic function of $t$ (refer Appendix). The heuristic function is the sum squared error function is given in equation (7). The error function is quartic rational function in $t$. The minimums are of the roots of the derivative of the Error function (refer equation (8)); the roots of the derivative are exactly the roots of its numerator, since the denominator is finite with the exception of $\pm\infty$. The numerator is a fifth degree polynomial equation, and its roots are found numerically by finding the eigenvalues of the corresponding companion matrix[14].

$(\mathcal{R}_0,\ \mathcal{R})$ and verify that pose relative to the rest of the scanline data. We only need to compute $\mathcal{PPE}$ sets
($\{\ \mathcal{PPE}\ (\Delta=|\ \mathcal{R}_0|\ \pm\ \epsilon\ ],\ \theta\ +\angle\mathcal{R}\mathcal{PPE}\ (\Delta=|\ \mathcal{R}_1|\ \pm\ \epsilon\ ],\ \theta\ +\angle\mathcal{R}$.

The *e ffi c i e n t* algorithm for computing $\mathcal{PPE}\ (\delta\ )$ involves two steps: enumerating all of the continuous families
of chords, and then enumerating all of the chords within each family. Orientations corresponding to continuous
family of ranges can be computed by scanning through the LongestChord function, searching for orientation
ranges which satisfy $\delta\ <\ \mathcal{LC}_{O}(\theta\ )$. After finding an orientation $\mathcal{LC}_{O}(\theta\ )=\delta$ which is the extreme chord in a
continuous family of chords, all of the chords in the family can be enumerated in two steps: rotating the *c h o r d*
*a b o v e  t h e  L o n g e s t  Ch o r d* counterclockwise, and rotating the *c h o r d  b e l o w  t h e  L o n g e s t  Ch o r d* clockwise.

### 3.1.1   Computing $\mathcal{PPE}\ (\Delta\ O)$

This section describes an efficient algorithm for computing $\mathcal{PPE}\ ([\delta\ ],\ O)$. First, an extremal chord from continuous
families of $\delta$-chords are found by using the longest chord function, and then the pairs of edges which contact
this family of chords are enumerated by walking the chord around the boundary. This construction computes
$\mathcal{PPE}\ ([\delta\ ],\ O)$ in $O(n\ +p)$ time where $n$ is $O$'s complexity and $p$ is the size of $\mathcal{PPE}\ ([\delta\ ],\ O)$, as follows:

1. Compute an extremal chord $c_{initial}$ of each continuous family of chords by searching $O$'s longest chord
   function for $\theta$ such that $\mathcal{LC}\ (O,\theta_{extremal})\ =\delta$ and referencing ExtremalVertex$(\theta_{extremal})$ and
   ExtremalEdge$(\theta_{extremal})$.

2. Beginning with $c_{initial}$, enumerate all of the chords in $c_{above}$ by walking $c_{initial}$ counterclockwise, and
   enumerate all of the chords in $c_{below}$ by walking clockwise. If $\delta$ was so short that Chords$(\Delta=[\delta\ ],\ \Theta,\ \overline{O})$
   is a single continuous family, we only need to walk $c_{above}$ around the boundary. Collect the edge-edge pairs
   contacted by both $c_{above}$ and $c_{below}$. Rotate the chords until:

   (a) $\mathcal{LC}_{O}(\theta_{c_{rotate}})\ =\delta$; the chord cannot be rotated any more.
   (b) $c_{rotate}\ =c_{initial}$ (the chord swept completely around $O$)

   These pairs of edges constitute the subset of $\mathcal{PPE}$ in that continuous family of chords.

### 3.2   Merging $\mathcal{PPE}$ sets

In this section, we describe the procedure for generating all feasible matches between two orientationally sorted
$\mathcal{PPE}$ sets. There are two main ideas of the *m e r g e* subroutine: the first idea is that only edge-edge pairs with
consistent *b o t t o m e d g e s* (containing the scanline origin) are checked, and the second idea is that only pairs of
edges with consistent orientations are checked (by scanning down $\theta$ ). The algorithm is detailed below

1. Enumerate $\theta$ from 0 to $2\pi$

2. $\forall$ edge-edge pair combinations of $\{\ \mathcal{PPE}\ (\Delta=|\ \mathcal{R}_0|\ \pm\ \epsilon\ ]),\ \theta\ +\angle\mathcal{R}\times\ \mathcal{PPE}\ (\Delta=|\ \mathcal{R}_1|\ \pm\ \epsilon\ ]),\ \theta\ +\angle\mathcal{R}\times\ .\ .\ .\ \}$
   with consistent *b o t t o m e d g e s* (scanline origin edges):

   - For all combinations of edges $\in\ \{\ \mathcal{PPE}\ (\Delta=|\ \mathcal{R}_0|\ \pm\ \epsilon\ ],\ \theta\ +\angle\mathcal{R}\mathcal{PPE}\ (\Delta=|\ \mathcal{R}_1|\ \pm\ \epsilon\ ],\ \theta\ +\angle\mathcal{R},\ .\ .\ .\ \}$,
     test whether each combination/interpretation is consistent (refer section 3.3). Each combination (
     $\mathcal{PPE}_{i_0,j_0}\ \times\ \mathcal{PPE}_{i_1,j_1}\ \times\ .\ .\ .$ ) refers to a candidate set of edges ($i_0,\ i,\ .\ .\ _0,\ ,j,j\ .\ .\ .$ ) which correspond
     to the relative chords of the scanlines. If any of the $\mathcal{PPE}_i$ is empty for some orientation $\theta\ +\angle\mathcal{R}_i$,
     then there is definitely no consistent set of edges. If the interpretation is consistent, then it is a valid
     correspondence.

### 3.3   Determining consistent correspondences algebraically

Given a hypothetical interpretation of edge-edge-pairs, the next problem is to determine whether the interpre-
tations are consistent. Although we could compute the optimal pose based upon those correspondences, this may
take too much time, and we would rather use a quick, but, perhaps less than perfect test. Since all of the relative
chords are anchored at the same scanline origin, all of the chords should share the same "bottom" edge.

Given *p e r f e c t* data and the correct correspondences, the correct pose should map all of the "bottom" points
of the relative chords onto the same scanline origin. This suggests a less sophisticated approach which involves
checking the distances between all of the "bottom" points, and computing the orientation $\theta$ which minimizes that
distance.

Many small edges
near the **center**
of the circle

distance
approxmiately
**radius**
of the circle

Many large edges approximating the
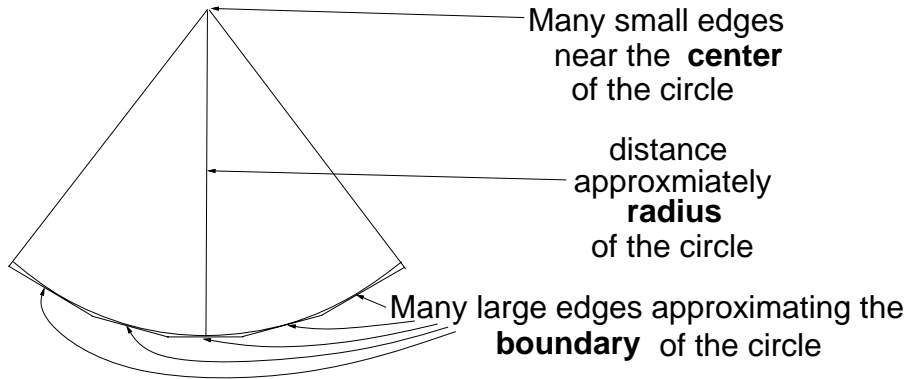**boundary** of the circle

Figure 15: In the worst case, as shown above, $| \mathcal{PPE} |$ can be $\frac{n^2+n}{2}$

## 3   CORRESPONDENCE PROBLEM

In this section, we give an overview of the correspondence algorithm, descriptions of the important subroutines,
and an analysis of the algorithm's time complexity.

### 3.1   Correspondence algorithm overview

The correspondence algorithm we present is an example of a *g e n e r a t e  a n d  t e s t* technique. The main idea of
the correspondence algorithm is to generate all the possible hypothetical interpretations for each relative chord
($\mathcal{R}$), and then test all combinations of these interpretations to find the correct interpretation. The hypotheses are
represented by $\mathcal{PPE}$ sets, which can be computed for arbitrary polygons in $O(n^2)$ time, or in $O(n+A)$ time for
convex polygons, for each chord. The correspondences are found by merging the $\mathcal{PPE}$ sets for all of the chords.
Each set contains edge-pairs and a range of angles for which that pair is feasible. In the convex case, only two
edge-pairs can occur for any fixed $\theta$. Thus, while merging two sets, at most 4 combinations can occur in each
range, and only two when the relative chord constraint is taken into account. In the non-convex case, there might
be $O(n^2)$ edge pairs for a single $\theta$, and $O(n)$ of them might be consistent with an edge-pair of a second $\mathcal{PPE}$ set.
The merging step can therefore take $O(n^3)$ steps in the general case.

1. For each relative chord $\mathcal{R}$, construct the sets of pairs of edges ($\mathcal{PPE}$ ($\Delta=[| \mathcal{R}_i | \pm \epsilon ]$)) for each of the relative
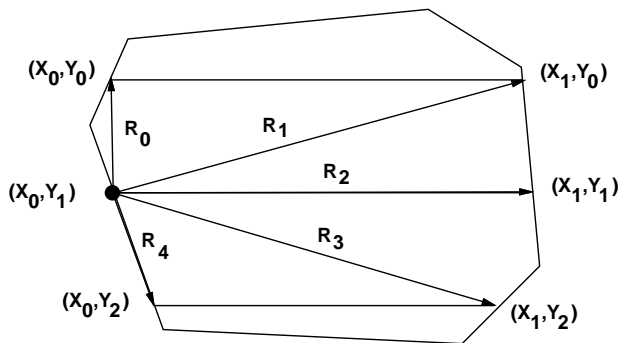   chord's distances (denoted by $R_i$ in Figure 16).



Figure 16: The relative chords between all of the scanline endpoints and the scanline origin are denoted by $\mathcal{R}_i$.
The lengths of the relative chords $| \mathcal{R}_i |$ constrain the contacting edges.

2. Merge the $\mathcal{PPE}$ sets in the manner described in section 3.2.

Since the scanline datapoints are very precise, we can optimistically assume that the relative chords constrain
the object's pose with high accuracy. In that case, we only need to compute the pose using two relative chords,

$$|Chords(O, \delta, \theta)| = \begin{cases} 0 & \text{if } \delta > \mathcal{LC}_O(\theta) \\ 1 & \text{if } \delta = \mathcal{LC}_O(\theta) \\ 2 & \text{if } \delta < \mathcal{LC}_O(\theta) \end{cases} \tag{4}$$

**Claim 4** *It is obvious that the positions of a $\delta$-chord's endpoints are continuous function $\mathcal{LC}_O(\theta)$.*

**Lemma 5** *$\mathcal{PPE}(\Delta)$ is computable in $O(A)$ time given the extremal chords of the chord families.*

**Proof** Consider walking the *a b o v e* and *b e l o w* chords around the boundary of $O$. As the orientation $\theta$ varies, the chords smoothly switch contact features. □

**Claim 6** *There are two continuous families of $\delta$-chords in every range $\delta < \mathcal{LC}_O(\theta)$.*

**Lemma 7** *When $\Delta_{max} < \mathcal{LC}(\theta)$, all of the chords $\in Chords(\Delta, [\theta, \overline{O}])$ ("above" the longest chord) lie between $Chords([\Delta_{min}], [\theta, \overline{O}])$ and $Chords([\Delta_{max}], [\theta, \overline{O}])$*

**Proof** Pictorial proof (refer Figure 14).
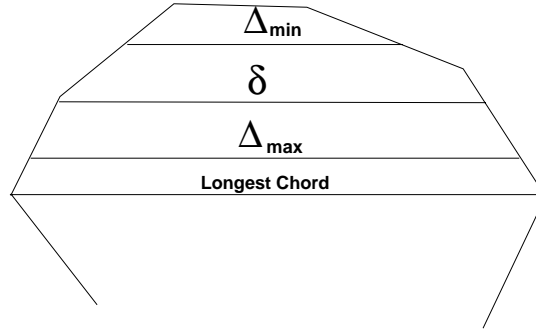


Figure 14: All of the chords $\in Chords(\Delta, [\theta], \overline{O})$ ("above" the longest chord) lie between $Chords([\Delta_{min}], [\theta, \overline{O}])$ and $Chords([\Delta_{max}], [\theta, \overline{O}])$

**Claim 8** *$\mathcal{PPE}(\Delta)$ for a particular orientation $\theta$ is generated by pairing up all of the edges to "edges" of $\mathcal{PPE}([\delta_{min}])(\theta)$ and $\mathcal{PPE}([\delta_{max}])(\theta)$ the edges between the "top edges" of $\mathcal{PPE}([\delta_{min}])(\theta)$ and $\mathcal{PPE}([\delta_{max}])(\theta)$.*

**Claim 9** *$\mathcal{PPE}(\Delta)$ can be conservatively estimated by computing $\mathcal{PPE}([\delta_{min}])$ and $\mathcal{PPE}([\delta_{max}])$, and then scanning down $\theta$ and enumerating all of pairs of edges in $\mathcal{PPE}([\delta_{min}])(\theta)$ and $\mathcal{PPE}([\delta_{max}])(\theta)$.*

**Lemma 10** *In the worst case there are at most $n^2$ ranges of orientations with different $\mathcal{PPE}$ sets.*

**Proof** The worst case bound $|\mathcal{PPE}| = n^2$ only occurs for a few scanlines even on the most pathological objects. An example of a worst case object is a pie wedge shaped object with lots of tiny edges at the tip and lots of long edges along the arc; there are a quadratic number of pairs of edges which accommodate chords of a length slightly smaller than the radial distance (refer Figure 15). □

**Lemma 11** *Each continuous family of chords includes at most $4n$ pairs of contact edges.*

**Proof** We prove a $2n$ bound assuming each edge is specifically a top or bottom edge, and $4n$ results from removing this constraint. Assume a chord is rotating through the family; let the terms *s w i t c h* describe the situation when one of the chord's endpoints changes its contacting feature. Except for the first and last switches, at least one of its endpoints completely sweeps over one of the edges at every switch. Each edge can support at most two directional families of chords, which implies that each edge can be completely swept over at most twice. Therefore, only $2n$ switches can occur. □

**Proof** This is shown by noting that if we translate the polygon in the direction of one of the longest chords, the displaced polygon intersects the original exactly at the point $v$, which implies that this is the largest chord in that direction. □
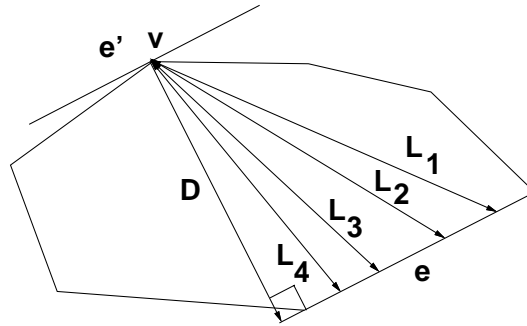


Figure 12: $L_1$, $L_2$, $L_3$, $L_4$ are longest chords for various orientations in the object shown above. For convex objects, when a longest chord contacts an edge $e$ and a vertex $v$, then all of the chords from the vertex $v$ to points along that edge $e$ are longest chords.

**Observation 2** *Every chord of O, being a difference between two boundary points of O, is a Minkowski sum $O + (-O)$, and the longest chords are points on the boundary of the Minkowski s*

Because of this observation, the longest chord function can be computed in $O(n)$ time. The Minkowski sum takes linear time to compute, and longest chord function, which is simply the radius function of $O + (-O)$ can be computed in linear time.

The length of chord of orientation $\theta$ contacting a particular edge $e$ an vertex $v$ is given in equation (3) where $D$ is the perpendicular vector from to $v$.

$$|L| = |D| \sec(\theta - \angle D) \tag{3}$$

**Lemma 3**

$$\forall O, \theta, \delta \quad Convex(O) \Rightarrow | Chords(\Delta = \delta, \Theta = \theta, Bound(O)) | \leq 2 s =($$
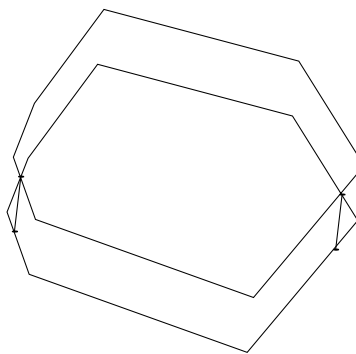


Figure 13: $\forall$ convex objects, there are at most 2 chords of length $\delta$ and orientation $\theta$

**Proof** Treat the chord $(\delta, \theta)$ as a translation vector, and the intersections between $\overline{O}$ and $\overline{O'}$ are the chords endpoints on $\overline{O}$. Since $O$ and $O'$ are both convex, there are at most 2 such intersections.

In general, there are either 0 or 2 chords of specified length and direction (refer equation (4)). For fixed orientations, the chordal distance decreases as one travels away from the longest chord; the 2 chords are found on opposite sides of the longest chord. □
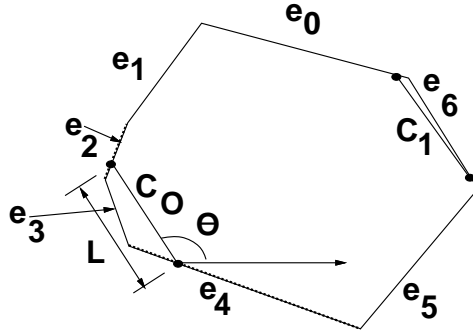
Figure 10: Depicted above are two chords $C_0$ and $C_1$ on an object's boundary. $\{C_0, C_1\} \in$ Chords($\Delta=[L]$, $\Theta=[\theta]$, $Boundary = \{e_0, e_1, \ldots e_6\}$,). EdgeEdgePairs($\{C_0\}$) = $\{(e_2, e_4)\}$



— CHORD

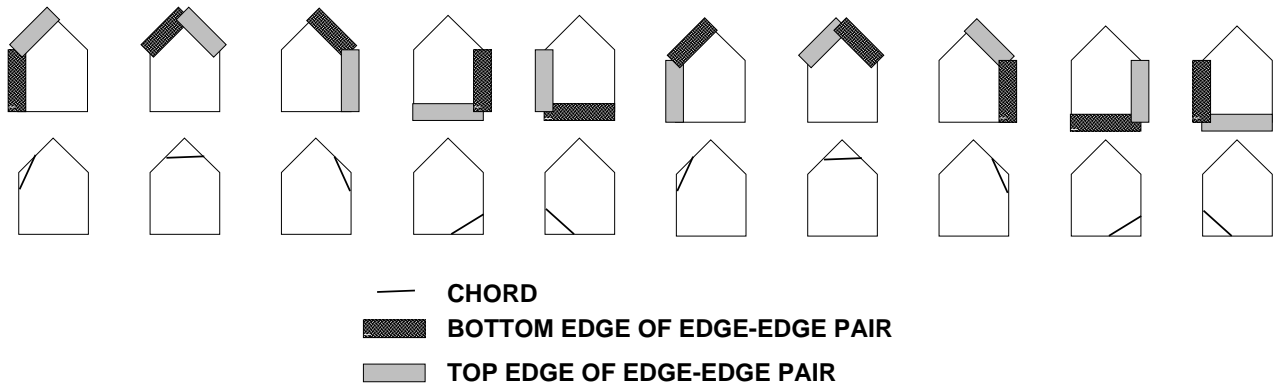▨ BOTTOM EDGE OF EDGE-EDGE PAIR

▭ TOP EDGE OF EDGE-EDGE PAIR

Figure 11: The set of edge-edge pairs for a fixed length chord and a particular convex object.

- The term $\mathcal{PPE}([\delta], O)$ is used to describe all of the *possible pairs of edges* ($\mathcal{PPE}$) which possibly contact a chord of length $[\delta]$ on the boundary of object $O$ (refer equation (2)); the chord's orientation is not constrained.

$$\mathcal{PPE}(\Delta, O) = Edge\,Edge\,Pairs(Chords(\Delta, \Theta=[\overline{0}, 2\pi]),$$ (2)

The brute force approach for computing $\mathcal{PPE}$ is to enumerate all pairs of edges; this takes $\frac{n(n-1)}{2}$ time.

## 2.2 $\mathcal{PPE}$ sets

The main idea of the *efficient* $O(n+A)$ correspondence algorithm is that $\mathcal{PPE}(\cdot)$, the pairs of possible edges, is computed in $O(n+A)$ time, where $n$ is the object's complexity and $A$ is the size of $\mathcal{PPE}(\cdot)$. The brute force approach of testing all pairs of edges takes $O(n^2)$ time.

The proofs in this section are limited to convex polygons; we prove that there are most two chords for particular length and orientation $\theta$. It follows that there could be $2^{2k-1}$ combinations of hypothetical chords for each $\theta$, where $k$ is the number of scanlines. But since we are using relative chords, which must all share the same endpoint, there are always at most 2 consistent combinations, independent of $k$. Checking these for particular $\theta$ takes constant time (treating $k$ as a constant), which gives us an overall running time of $O(n+A)$.

We also prove that the chords are arranged in continuous families and that there are at most $n$ such families of chords, where each family contains chords with at most $4n$ edge contact pairs.

**Lemma 1** *Let $e$ and $v$ be an edge and a vertex respectively of a convex object, with $v$ not in the the orientation of $e$ lies between the orientations of the edges that $v$ bounds, then all of $v$ to points along that edge $e$ are longest chords (refer Figure 12).*
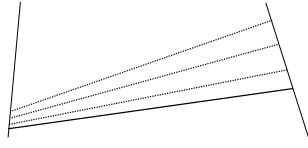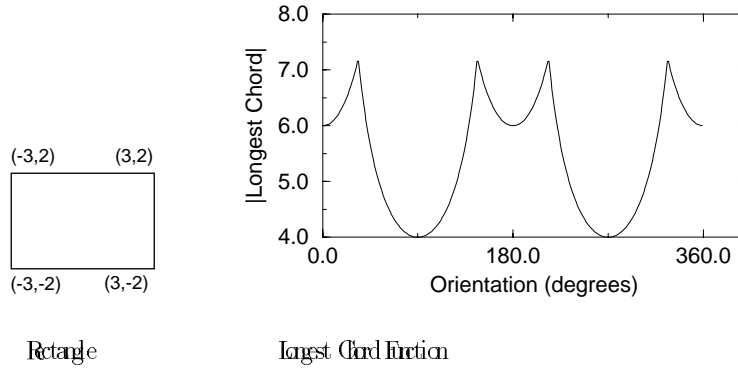
Figure 7: A chord rotating around two edges.



(-3,2)  (3,2)

(-3,-2)  (3,-2)

Rectangle                          Longest Chord Function

Figure 8: Longest Chord Function of a 2:3 Rectangle.

- ShortChordVertex($\delta$) and ShortChordEdge($\theta$) denote a vertex and edge which contact a $\delta$ length chord. They are used to find an initial vertex for computing $\mathcal{PPE}$ by walking around the object's boundary. We can compute ShortChordVertex by starting with any vertex $v$ and walking along the boundary $e_1, e_2, \ldots,$ recording the chord lengths between $v$ and $e_i$. The distances between $v$ and $v_{i+1}, v_i$ define the chord lengths between $e_i$ and $v$.
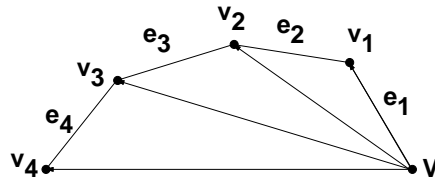


Figure 9: ShortChordVertex($\delta$) and ShortChordEdge($\theta$) are generated by starting with an arbitrary vertex $v$ and computing the distance ranges between $v$ and $e_i$. The distances between $v$ and $v_{i+1}, v_i$ define the chord lengths between $e_i$ and $v$.

- The range of values in the range between $x - \epsilon$ and $x + \epsilon$ is represented by the expression $[x - \epsilon, x + \epsilon]$. Exact values are represented by the expression $[x]$. The range is denoted by $[x \pm \epsilon]$.

- Chords($\Delta = [\delta_{min}, \delta_{max}]$, $\Theta = [\theta_{min}, \theta_{max}]$, $Boundary = \{e_0, e_1, \ldots\}$) refers the set of all chords of length $\delta$ such that $\delta_{min} < \delta < \delta_{max}$ and of angular inclination $\theta$ such that $\theta_{min} < \theta < \theta_{max}$ and both endpoints of the chords lie on one of the boundary edges in the set $e_0, e_1, \ldots$ (refer Figure 21).

- The term AngleRanges($\{c_0, c_1, \ldots\}$) is used to describe the set of orientations of all of the chords $\{c_0, c_1, \ldots\}$; the symbol $\angle$ is used as a shorthand for AngleRanges.

- The term EdgeEdgePairs($\{c_0, c_1, \ldots\}$) is used to describe all of the pairs of contact edges of all of the chords $\{c_0, c_1, \ldots\}$ (refer Figure 11). The terms "bottom edge" and "top edge" refer to the two edges of the edge-edge-pair. $bottomEdge_{chord}$, $topEdge_{chord}$ are shorthand for these terms.

- The term *c h o r d s* refers to lines connecting points on an object's boundary. All scan lines are chords (refer Figure 2.1). Chords are denoted by $c_i$.
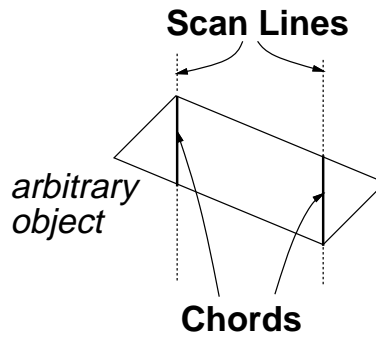


**Scan Lines**

*arbitrary object*

**Chords**

Figure 5: Chords generated by scanning an arbitrary object

- $Left_i(c_i, c_j)$ denotes a *l e f t - p r e d i c a t e* of two chords which is true if $c_j$ is entirely to the "left" of the directional line through $c_i$. A vector $v_i$ is left of a vector $v_j$ if the cross product $v_i \times v_j$ is positive.

- $\angle c_i$ denotes the angle included between the directional line through $c_i$ and the x axis.

- Chords $c_i$, $c_j$ are totally ordered by the following inequality:

$$c_i < c_j \Leftrightarrow (\angle(c_i) < \angle(c_j)) \vee (\angle(c_i) = \angle(c_j) \wedge Left_i(c_i, c_j))  \tag{1}$$

- The terms *s c a n l i n e   o r i g i n* refers to a specific scanline endpoint (refer Figure 6).



**Relative chords**
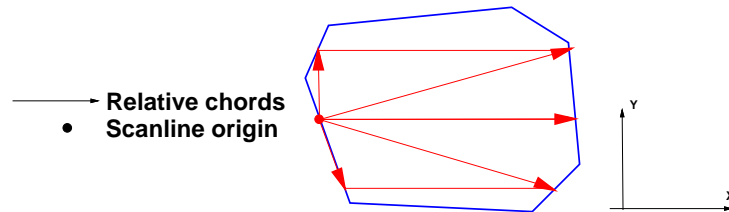• **Scanline origin**

Y

X

Figure 6: The *s c a n l i n e   o r i g i n* is a particular scanline endpoint, and the other scanline endpoints define chords relative to the origin

- The term $\delta$-chord is shorthand for a chord of length $\delta$ and unspecified orientation.

- The term *r e l a t i v e   c h o r d* refers to the chord between a scanline endpoint and the scanline origin. $2k - 1$ independent relative chords are obtained by scanning an object with $k$ scanlines. Relative chords are denoted by the symbol $\mathcal{R}$.

- A chord *r o t a t e s* around a boundary by as its orientation varies maintaining contact along the object's boundary. In general the chord's top and bottom contact points are continuously parameterized by orientation. The exception occurs when both endpoints touch parallel sides of the polygon (refer Figure 7).

- $\mathcal{LC}(\theta)$ is defined as the *l o n g e s t   c h o r d* of orientation $\theta$ on an object's boundary. See Figure 12.

- ExtremalVertex($\theta$) and ExtremalEdge($\theta$) denote the vertex and edge which contact LongestChord($\theta$). There are only $O(n)$ different extremal vertices and extremal edges (refer Observation 2).
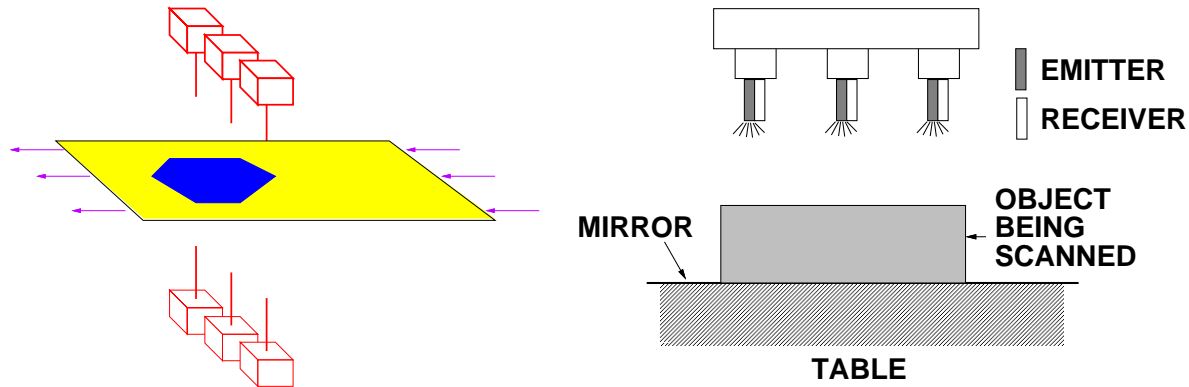
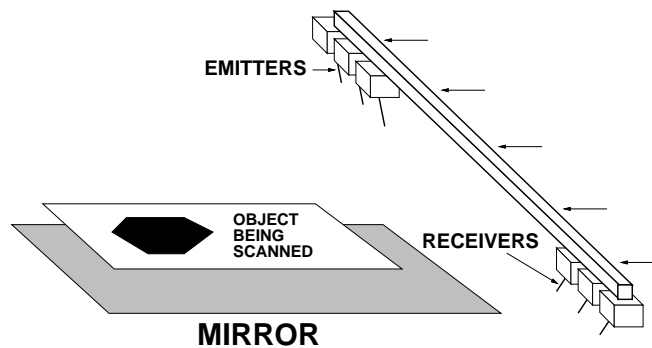Figure 3: Two abstract devices for scanning an arbitrary object.



Figure 4: A specialized scanning device for flat objects.

in practice). In the future, we plan to experiment with the sensor on a variety of industrial parts to see if this problem arises. If it does we plan to explore active beam sensing, where the beam is moved to a position on the boundary that will discriminate between confusible parts.

### 1.4 Outline

The rest of the paper is organized in the following manner. In section 2, we present the theoretical background, consisting of notation and the geometrical framework for the correspondence algorithm. In section 3, we present the correspondence algorithm for convex polygons, and we detail the algorithm for non-convex objects in section 4. In section 5, we describe a simple example. Finally, in section 6, we highlight the results and advantages of this technique.

## 2   THEORETICAL FRAMEWORK

This section describes the notation and theoretical background used in the correspondence algorithm. This technique assumes that the object is initially lying on a flat horizontal surface of its convex hull.

### 2.1 Notation

- $\mathcal{SH}(P)$ refers to the shadow of $P$, a polyhedral object, which is a vertical projection of $P$ onto the x-y plane (assuming the scanning beams are vertical). $\mathcal{SH}_f(P)$ is defined as the shadow of an object, $P$ resting on face $f$.

- The symbol $O$ is used to denote a two-dimensional shadow $\mathcal{SH}(P)$ of a polyhedron $P$.

- $\overline{O}$ refers to the boundary of a shadow $O$.

- A pair of vertices $v_i, y \in O$ denoted by $< v_i, y >$, is $extremal$ if the distance between the two vertices is locally maximized.

Related work has also been done in non-vision recognition and localization, which have been geometrically-based. Using finger gap distance measurements, Rao and Goldberg [10,11] automatically oriented parts by generating plans consisting of a series of oriented grasps. Wallack et al. [12] presented a recognition and localization technique using cross beam sensor measurements (repeatable to a few thousandths of an inch), but this technique was limited to non-flat objects and could not distinguish between objects whose cross-sections have identical convex hulls. The technique presented in this paper is directly related to Mroch et al.'s recognition and localization technique for scanline data which utilized a preprocessed lookup table [13].

The technique described in this paper is linked to the RISC (Reduced Intricacy in Sensing and Control) paradigm presented by Canny and Goldberg [1]. The RISC approach seeks to achieve intelligent flexible behavior by combining simple sensing and actuation elements. This report describes a correspondence algorithm for data from linear arrays of a few beam sensors, and we show that these sensors can recognize and localize objects to a few thousandths of an inch.

In the scope of this paper, we do not address the related problem of estimating the optimal pose given the data points and correspondences/feature interpretations. The optimal pose estimation problem is usually treated as a non-linear least squares problem solved either by the normal approach of iterative methods or by exact algebraic elimination theory as described by Wallack et al. [12].

## 1.2 Overview

This report describes an algorithmic solution to the scanline data correspondence problem. Scanline data is generated by moving scanning light beams or reflective sensors with respect to a three-dimensional object, and recording the sensors' positions at each breakpoint, when any of the sensor's outputs change. The breakpoints correspond to scanline endpoints. In this paper, we describe a technique for matching the scanline data to features of a modelled object (refer Figure 2): given a set of possible objects, the high precision data should be self-consistent with only a single interpretation of a single model.

Scanline data can be generated from a number of different methods: filtered image data, thresholded rangefinder data, laser based sensors, or customized light sensors. A transparent horizontal supporting surface is necessary in order to use *through beam* light sensors. Figures 3 and 4 depict different possible scanning apparatus. The simplest sensor design involves using reflective light sensors. For reflective sensors, the emitter and receiver pair should be at a specified distance away from the object.
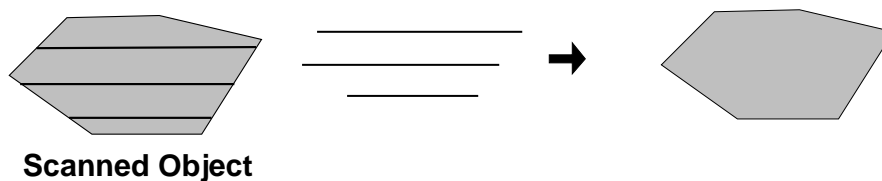


**Scanned Object**

Figure 2: Given the scan lines determined from an arbitrary object, the algorithm determines the object's identity and pose

In order to recognize three-dimensional objects using only a few high-precision data points, two inescapable assumptions were made:

1. In a controlled environment, such as manufacturing, parts are presented singly (singulated) and occlusion does not occur.

2. The polyhedral objects to be recognized are resting flat on a horizontal surface.

As a consequence of these assumptions, the correspondence problem is reduced to solving the correspondence problem for all possible shadows of a polyhedron

## 1.3 Foreword

As a result of the scanline sensing, the sensors only perceive a shadow of the object cast parallel to the beam. The technique cannot discriminate between two objects with the same shadow and fails in this case (binary vision has the same limitations). The technique may confuse objects with different shadows because the scanline sensor only registers a small number of points on the object's boundary (we do not know yet if this is a problem
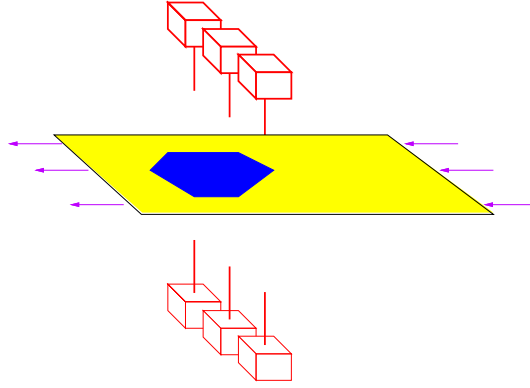
Figure 1: The scanning procedure.

In this paper, we describe an $O(n + A)$ correspondence algorithm for convex three-dimensional objects with polygonal silhouettes, and an $O(n^2 \log n + A)$ algorithm for non-convex three-dimensional objects, where $n$ is the object's complexity and $A$ is the total number of feasible matches. Typically for convex objects, the total number of matches is $O(n)$, but in the worst case it is $O(n^2)$. For non-convex objects, there are at most $O(n^3)$ matches.

## 1.1 Previous work

The correspondence problem for which we describe an algorithm for scanline data, has been extensively studied in the machine vision literature and many solutions to this problem have been proposed. The main difference between the technique which we present, and the majority of machine vision techniques in the literature, is that we are focusing on the problem of recognizing objects from sparse data, whereas most machine vision attempts to recognize objects from data-rich images. Another difficulty is that the probe data does not provide information about the contact features (edges and vertices) such as the normal vector which would be accessible from camera data. The advantage of using simple sensors is their high precision and repeatability. The sensor's accuracy enables quick recognition and localization algorithms.

The correspondence problem can be described as an interpretation problem the task is to find all valid interpretations of the data with respect to the model, or, alternatively, to enumerate all self-consistent matches between data features and model features. In essence, this problem involves a combinatorial search; without an efficient heuristic for pruning inconsistent interpretations, determining the correspondences would take time exponential in the size of the matching sets.

Much work has been focused on solving the correspondence problem in a tractable amount of time, quickly enough for real-time systems. One approach from the literature involves generating transform hypotheses by matching tuples of data features to tuples of model features and then corroborating or dismissing each transform hypothesis using the other data features. Ayache and Faugeras presented the HYPER system[3] to recursively match edges from the image to edges in the model. A refinement of this method called the *alignment method* was studied by Huttenlocher and Ullman[4] who used carefully chosen sets of features to compute the transform. Kalvin, Schonberg, Schwartz and Sharir described an object recognition technique which utilized pose-independent features called footprints (characteristic boundary curves) to recognize planar objects, and their technique succeeded even when objects were occluded[5]. Lamdan and Wolfson[6] introduced geometric hashing as an efficient method for recognizing objects by utilizing precomputed hash tables of sensor values.

The general approach of using precomputed tables is termed *indexing*, Clemens and Jacobs[7] thoroughly discuss this field. Another type of precomputation approach involves *interpretation trees*. This reduces the correspondence problem to a tree-search problem where incompatible interpretations have been pruned offline. Grimson[8] analyzed interpretation trees and showed an expected polynomial bound on the number of search steps with the dimension depending on the number of features required to compute the transform.

Another approach to solving the correspondence problem combines indexing with geometric pose invariant properties, which remain invariant with respect to pose and viewposition. The advantage is that each correspondence tuple has a single entry. Forsyth et al. analyze invariants and present a curve based object recognition and localization technique[9].

# A geometric matching algorithm for beamscanning

Aaron S. Wallack*    John F. Canny [†]

Computer Science Division

University of California

Berkeley, CA 94720

## Abstract

In this paper, we present a model-based object recognition technique using scanline information. Objects are scanned using a small number of on/off light sensors. The times when the beams break and reconnect constrain the object's identity, position, and orientation. We study this type of sensor because these sensors are inexpensive, compact, precise, insensitive to ambient light, and well-suited to manufacturing environments. The sensor provides sparse information consisting of isolated points on the object's boundary, and does not provide normal information. Conventional model-based matching techniques, such as the alignment method, take $O(n^3)$ time to solve this problem. We describe an $O(A+n)$ correspondence algorithm for objects with convex polygonal silhouettes, where $n$ is the silhouette's complexity, and $A$ is the total number of consistent edge pair matches for pairs of scanline points. The total number of edge pair matches is $O(n^2)$ in the worst case, but typically $O(n)$. Our algorithm also works for non-convex objects, but the number of edge pair matches is typically somewhat larger $O(n^2)$ in the worst case, $O(n^3)$. The reason that we focus on the correspondence problem is that given the correspondence information, the object's position and orientation can be easily computed from the data points and corresponding features.

## 1  INTRODUCTION

We study the problem of model-based object recognition from sparse data. The data consist of points on the objects silhouette or shadow obtained from an array of light beam sensors as shown in Figure 1. This research is part of a larger effort termed RISC Robotics [1] (Reduced Intricacy Sensing and Control) which attempts to solve manufacturing problems with simple hardware and sophisticated software. In a nutshell, beam sensors are well-suited for manufacturing because they provide very high accuracy and reliability at low cost, they are small enough to use in tight spaces, and their high accuracy enables simple, fast recognition algorithms. This kind of recognition is related to the *s h a p e - f r o m - p r o b i n g* problems described by Cole and Yap [2], which determine an object's pose and even shape from a small number of boundary points.

It is difficult to implement fast recognition algorithms because of the sparse data, in particular the absence of normal information. One of the major subtasks of object recognition is the *c o r r e s p o n d e n c e   p r o b l e m*, the problem of matching data points to model features. Typically we have only six points on the object's boundary and no clue as to which edges they correspond. Each generic match between a data point and a model feature provides one constraint. We assume that the object is stably resting on a flat surface, and therefore we only need to consider two dimensional silhouettes (with three degrees of freedom). Thereby, three generic matches are sufficient to constrain the object's pose; *e x t r a n e o u s* matches are used to verify poses by estimating the poses credibility. Model-based matching techniques, such as the alignment method, would require at least $O(n^3)$ steps to do this.