

FINDING CYLINDERS IN RANGE DATA

Tomás Lozano-Pérez
W. Eric L. Grimson

MIT Artificial Intelligence Laboratory
Cambridge, MA 02139

Steven J. White
Technical Arts Corporation
Redmond, WA 98052

Abstract. We have investigated the problem of locating cylinders in a depth scan map. The crucial problem is deciding whether a group of scans could arise from the same cylinder. We investigated a number of traditional approaches to this problem. We found significant reliability and accuracy problems in the traditional approaches that involve fitting ellipses to the scan data. As an alternative, we have developed a simple and very robust method for computing the axis of a cylinder based on three scans. This computation provides the basic capability needed to segment the scan data. This report summarizes our experience with several of the methods and describes the new method in detail.

1. The problem

The problem we address in this paper is that of locating cylinders in data from a light-stripe based range-finder. We assume that we have available a number of *scans* which represent the distance from the sensor, that is, *depth*, of the nearest points on the intersection of a light plane and the scene (Figure 1).

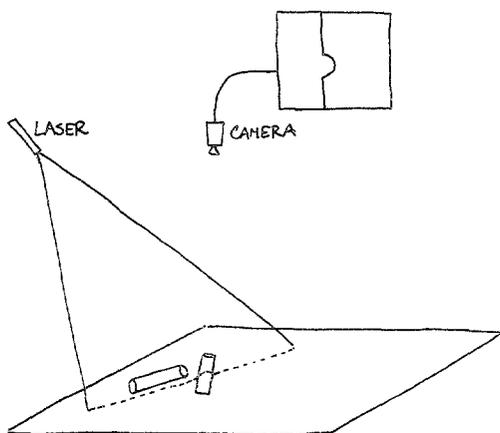


Figure 1. Schematic of laser sensing system. For each position of the plane of light, a scan of points in the camera can be processed to determine three-dimensional position.

Our goal is to identify the presence of cylinders in this data. We assume that this is to be done by hypothesis accumulation, that is, by testing one or more scans to construct a hypothesis for a cylinder axis and, then, identifying large consistent sets of hypotheses. Therefore, the crucial step in the computation is constructing a cylinder axis from one or more scans. The traditional approach to this problem is to fit an ellipse to the scan. The parameters of the ellipse can be used to solve for the cylinder axis [Bolles 86]. The advantages of this technique are that only one scan is required to construct a cylinder hypothesis and that the solution is independent of sensor location. The fitting technique, however, is subject to numerical accuracy problems, especially when only small segments of the ellipse are available.

In this paper, we develop a very simple technique for solving for the orientation of the cylinder axis given three scans on the cylinder. The technique is very stable numerically. In addition, like ellipse fitting, the solution is independent of sensor location.

2. Our Approach

2.1 Characterizing the cylinder axis

The method we use for this task is based on the following geometric question (refer to Figure 2):

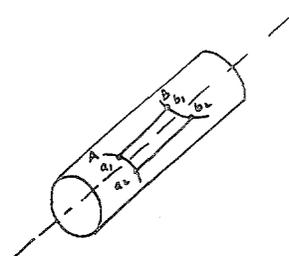


Figure 2. Two scans A and B on a cylinder. The question is whether the line segments a_1b_1 and a_2b_2 are parallel only when they are also parallel to the cylinder axis.

Take two scans (A and B) on a cylinder, obtained as the intersection of two different light planes with the cylinder. Pick two pairs of points on these scans: a_1, a_2 and b_1, b_2 . Under what conditions are the line segments $\overline{a_1 b_1}$ and $\overline{a_2 b_2}$ parallel. In particular, what additional conditions will imply that they are parallel only when they are also parallel to the cylinder axis.

To answer this question, we need to determine the conditions under which we can find four points on the surface of a cylinder, such that the lines between two pairs of points are parallel. Without loss of generality, we can consider a cylinder of radius r , with axis oriented along the \hat{z} axis, and centered at the origin. A point on the cylinder is given by

$$\mathbf{p}(\theta, h) = (r \cos \theta, r \sin \theta, h)$$

Without loss of generality, we can incorporate r into the height h , by scaling the dimensions of the axes. In this case, points on the cylinder are given by

$$\mathbf{p}_i(\theta, h) = (\cos \theta_i, \sin \theta_i, H_i) \quad i = 1, 2, \dots$$

Now we are free to rotate the coordinate system about the \hat{z} axis. Thus, without loss of generality, we choose a coordinate system in which

$$\theta_2 = -\theta_1.$$

Hence, our four points can be represented by

$$a_1 = \mathbf{p}_1 = (\cos \theta_1, \sin \theta_1, H_2)$$

$$b_1 = \mathbf{p}_2 = (\cos \theta_1, -\sin \theta_1, H_2)$$

$$a_2 = \mathbf{p}_3 = (\cos \theta_3, \sin \theta_3, H_3)$$

$$b_2 = \mathbf{p}_4 = (\cos \theta_4, \sin \theta_4, H_4)$$

Now consider the vectors $\mathbf{p}_1 - \mathbf{p}_2$ and $\mathbf{p}_3 - \mathbf{p}_4$

$$\mathbf{p}_1 - \mathbf{p}_2 = (0, 2 \sin \theta_1, H_1 - H_2)$$

$$\mathbf{p}_3 - \mathbf{p}_4 = (\cos \theta_3 - \cos \theta_4, \sin \theta_3 - \sin \theta_4, H_3 - H_4).$$

These vectors are parallel if and only if their components are equal, modulo the same scale factor, or equivalently, if their cross product is the zero vector. By looking at components, we arrive at three conditions on the vectors, in order for them to be parallel.

$$2 \sin \theta_1 (\cos \theta_3 - \cos \theta_4) = 0 \quad (1)$$

$$(\cos \theta_3 - \cos \theta_4) (H_1 - H_2) = 0 \quad (2)$$

$$2 \sin \theta_1 (H_3 - H_4) = (\sin \theta_3 - \sin \theta_4) (H_1 - H_2) \quad (3)$$

Now equation (1) implies either $\sin \theta_1 = 0$ or $\theta_4 = \pm \theta_3$. Equation (2) implies either $\theta_4 = \pm \theta_3$ or $H_1 = H_2$.

Thus, we can exhaustively examine the conditions needed for parallelism case by case.

1. Suppose $\theta_4 = \theta_3$. In this case, it is clear that $\mathbf{p}_3 - \mathbf{p}_4$ is parallel to \hat{z} , as desired.
2. Suppose $H_1 = H_2$ and $\sin \theta_1 = 0$. This is the trivial case of $\mathbf{p}_1 = \mathbf{p}_2$, which we will ignore.
3. Suppose $\theta_4 = -\theta_3$. In this case, we have

$$\mathbf{p}_1 - \mathbf{p}_2 = (0, 2 \sin \theta_1, H_1 - H_2)$$

$$\mathbf{p}_3 - \mathbf{p}_4 = (0, 2 \sin \theta_3, H_3 - H_4).$$

If $\sin \theta_1 = 0$ or $\sin \theta_3 = 0$ then we have the desired case of the lines being parallel to the \hat{z} axis. So the only case remaining is given by the above two equations under the condition that $\sin \theta_1 = 0$ and $\sin \theta_3 = 0$. Unfortunately, so long as

$$\sin \theta_1 (H_3 - H_4) = \sin \theta_3 (H_1 - H_2)$$

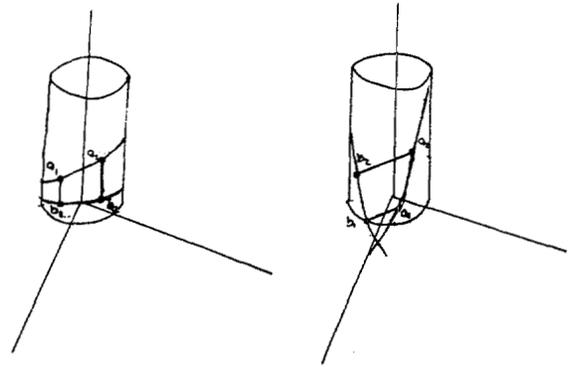


Figure 3. Two solutions to the problem of parallel lines, whose endpoints lie on the surface of the cylinder. In the case on the left, the lines are also parallel to the cylinder axis. In the case on the right, they are not.

we have a valid set of points, such that their separation vectors are parallel, but are not themselves parallel to the cylinder axis.

Two different classes of such solutions are shown in Figure 3. We need an additional condition that will allow us to distinguish the case on the right from that on the left. The figure allows us a simple observation. In the case on the right, the two lines will only intersect the cylinder at the chosen points. Elsewhere, the lines will either line inside or outside of the cylinder itself. On the other hand, in the case on the left, the entire lines lie along the surface of the cylinder. Thus, suppose we take a third slice of the cylinder, and further suppose that on that slice, we can find a point

$$\mathbf{p}_5 = (\cos \theta_5, \sin \theta_5, H_5)$$

such that $\mathbf{p}_1 - \mathbf{p}_5$ is parallel to $\mathbf{p}_1 - \mathbf{p}_2$. Then we have

$$\mathbf{p}_1 - \mathbf{p}_5 = (\cos \theta_1 - \cos \theta_5, \sin \theta_1 - \sin \theta_5, H_1 - H_5)$$

$$\mathbf{p}_1 - \mathbf{p}_2 = (0, 2 \sin \theta_1, H_1 - H_2)$$

Thus, these two lines being parallel implies $\theta_1 = \pm \theta_5$.

1. Suppose $\theta_1 = \theta_5$. Then clearly $\mathbf{p}_1 - \mathbf{p}_5$ and $\mathbf{p}_1 - \mathbf{p}_2$ are both parallel to $\hat{\mathbf{z}}$ and we are done.
2. Suppose $\theta_1 = -\theta_5$. Then

$$\mathbf{p}_1 - \mathbf{p}_5 = (0, 2 \sin \theta_1, H_1 - H_5).$$

Now, $\mathbf{p}_1 - \mathbf{p}_5$ and $\mathbf{p}_1 - \mathbf{p}_2$ being parallel implies either $\sin \theta_1 = 0$ or $H_1 - H_5 = H_1 - H_2$. But if $\mathbf{p}_1 = \mathbf{p}_5$, then the second case cannot be true, which implies $\sin \theta_1 = 0$, which implies that the lines are all parallel to the $\hat{\mathbf{z}}$ axis.

This allows us to make the following observation.

Take three scans (A, B and C) on a cylinder, obtained as the intersection of three different light planes with the cylinder. Pick two pairs of points on the first two scans: a_1, a_2 and b_1, b_2 . If the line segments $\overline{a_1 b_1}$ and $\overline{a_2 b_2}$ are parallel, and there is a point a_3 on the third scan, that lies on the extension of the ray $\overline{a_1 b_1}$, then the lines segments are also parallel to the cylinder axis.

2.2 Solving for the cylinder axis

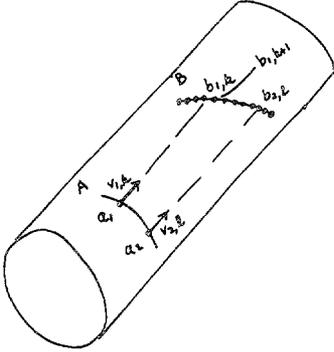


Figure 4. Illustration of a linear time algorithm for finding the cylinder axis.

Consider how this fact can be used to develop an algorithm for solving for the cylinder axis (Figure 4). Given two scans (A and B), pick two points on one scan, $a_i (i = 1, 2)$; the points should be widely separated on the scan, but they must lie on the common overlap of the two scans. We must choose $b_i (i = 1, 2)$ on the B scan. We will write $b_{i,k} (k \leq n)$ to indicate the choice of the k^{th} point on the B scan as b_i . Then, for each point $b_{i,k} (k \leq n)$ on the B scan, construct the unit vector $v_{i,k}$ from a_i pointing at the point $b_{i,k}$. The dot product $v_{1,k} \cdot v_{2,l}$ (for any choice of points $b_{1,k}$ and $b_{2,l}$) measures the cosine of the angle between the two line segments. The brute force algorithm simply finds all the dot products $v_{1,k} \cdot v_{2,l}$ and picks the one closest to 1.0. If there are n points on scan B this operation requires computing $2n$ unit vectors and performing n^2 dot products. But, we can do better.

A more efficient algorithm exploits what we know of the geometry. We start with $k = 0, l = 0$, that is, the initial

B points are $b_{1,0}$ and $b_{2,0}$; note that these are the *same* endpoint of the B scan. The basic loop of the algorithm increments k and then executes a loop that increments l and evaluates $v_{1,k} \cdot v_{2,l}$ until it finds a maximum. This loop is repeated until the value of the best dot product starts dropping.

```

 $l_{best} = 0$ 
 $dot_{best} = v_{1,0} \cdot v_{2,0}$ 
Loop for  $k$  from 0 to  $n$ 
   $dot = dot_{best}$ 
  Loop for  $l$  from  $l_{best}$  to  $n$ 
    If  $v_{1,k} \cdot v_{2,l} \geq dot$ 
      Then  $dot = v_{1,k} \cdot v_{2,l}; l_{best} = l$ 
    Else Exit Loop
  End Loop
  If  $dot \geq dot_{best}$ 
    Then  $dot_{best} = dot$ 
  Else Return( $v_{1,k-1}$ )
End Loop

```

The value of dot_{best} should approach 1.0 monotonically and then start to drop off. At the peak, both the $v_{i,k}$ are estimates of the cylinder axis direction. This estimate can be refined to sub-pixel resolution by interpolating between the points $b_{i,k}$ using, say, a circular interpolant. In the worst case this algorithm computes $2n$ unit vectors and n dot products. In fact, one expects significantly less than these bounds. If there is relatively little rotation of the scanner between the A and B scans, then the initial choice of k and l to be greater than 0 can significantly reduce the computation.

One important caveat in using the above algorithm is that the presence of noise can lead to false local extrema in the value of the dot product. Our implementation actually looks ahead on the scan a bit to make sure that the extremum is the global one.

It is desirable to average the results obtained from a variety of choices of the points a_i , to reduce the effects of error in measuring the position of the a_i . For example, given points a_1, a_2, a_3 , the operation described above can be repeated for the three different combinations (a_1, a_2) , (a_2, a_3) and (a_1, a_3) .

Once we have obtained an estimate of the cylinder axis direction from a pair of scans we need to incorporate a third scan to test that the estimated axis is in fact a real one. We could do this by extending one of the two parallel lines until it intersects with the plane of a third scan and testing that the intersection point also lies on the cylinder. Alternatively, we can cluster axes hypotheses from successive pairs of scans. If a sequence of scans pairwise give rise to roughly the same axis direction, then the condition of being parallel to the cylinder axis is automatically satisfied.

We have found that the axis direction is sufficient to do the grouping of cylindrical segments, but once the axis is known it is straightforward to compute numerically an estimate of the axis displacement. Any of the methods in section 3 will serve. In our testing, we have either simply used the position of the average point of the scan as a (very) rough estimate of the center of the cylinder, or we have used a straightforward Hough transform technique to determine the offset of cylinder axis.

We have tested this method in two ways:

- We have taken a number of very accurate scans of a number of cylinders and computed the axis based on that data. What we found is that we can estimate the direction of the cylinder axis to approximately 1 or 2 degrees on the basis of two scans taken 0.2 inches apart. For scans taken 1.0 inch apart, the estimate is less than 1 degree from the correct axis.
- We have performed a large number of experiments using much lower accuracy data with a variety of cylinders and other elongated objects (cables). This data gives us an idea of how robust the method is. We have seen no catastrophic failure even for noisy data taken from surfaces that are not truly cylindrical. The axis direction estimates we obtain from cylindrical segments tend to be clustered within 5 degrees. The method even tolerated scans that were partially occluded by overlapping objects. Figure 5 shows some scans from a representative tube and the linear segments identified by the algorithm. The position of the axis in these figures is not being computed accurately, only the axis direction is reliable.

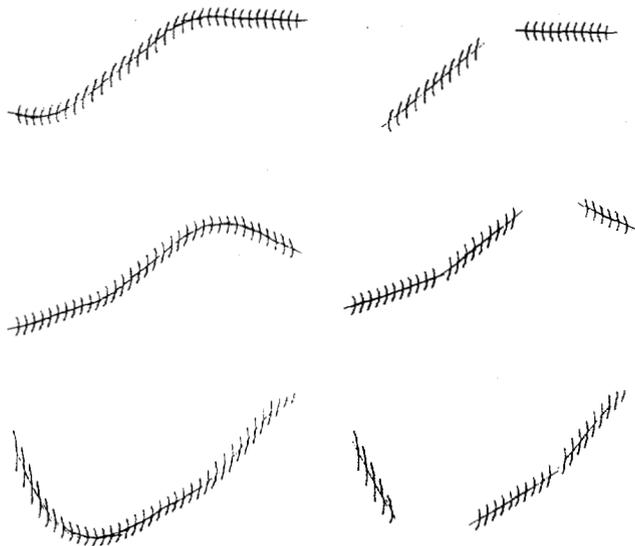


Figure 5. Examples of tube axes computed by the described method.

2.3 Why it works

Clearly, the key to the performance of the algorithm is the parallel-line observation of the previous section. The development of section 2.2 shows algebraically why this observation should hold, and the experiments show that it does, in fact, work as desired. In this section, we provide a geometric argument as to *why* it works. Consider the points a_1, a_2, b_1 chosen so that the line segment $\overline{a_1 b_1}$ is *not* parallel to the cylinder axis. The key question is: Can we choose b_2 so that $\overline{a_1 b_1}$ is parallel to $\overline{a_2 b_2}$?

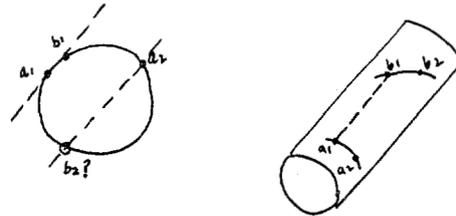


Figure 6. Part (a). End-on view of the scans on the cylinder with the chosen points indicated. Part (b). Side view of the same situation.

The simplest way of visualizing the situation is shown in Figure 6. This picture helps us see why we cannot generally pick b_2 to satisfy the conditions above. Imagine looking along the cylinder axis, then the points on all scans map into segments of the same circle (save for measurement error). We require that $\overline{a_1 b_1}$ be parallel to $\overline{a_2 b_2}$. Recall that parallel lines are always parallel under any rectilinear projection, therefore the lines $\overline{a_1 b_1}$ must be parallel to $\overline{a_2 b_2}$ in the end-on view of the cylinder. Note that only points along the cylinder axis map into the same point on the projection so, by construction, a_1 and b_1 are different points on the circle (Figure 6) and define a line in the projection. Since b_2 must be chosen to be on the circle, it must be at the intersection of the line parallel to $\overline{a_1 b_1}$ with the circle. This intersection will be on the opposite side of the circle from a_2 . If the scans are mostly overlapping in the projection, b_2 cannot be chosen to be on the B scan.

There is at least one apparent counterexample to the argument given above. What if the scans cross (see figure 7)? Then it should be possible to choose parallel lines as shown in the figure. Do these correspond to an axis? We believe that the answer is yes! If one can find two sets of parallel lines then the scans are truly ambiguous; there really are two cylinders that can give rise to that pair of scans. In fact, one can see that when the two crossing scans are perpendicular to each other, two sets of parallel lines exist and the situation is ambiguous. As the scans become more parallel, the ambiguity disappears (and only one set of parallel lines can be found). More on this in section 2.3.

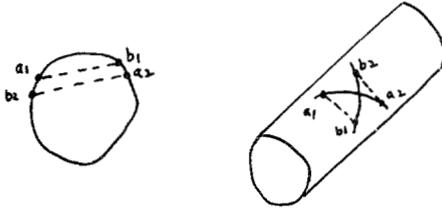


Figure 7. Part (a). End-on view of the scans on the cylinder with the chosen points indicated. Part (b). Side view of the same situation.

An alternative construction (Figure 8) reaches the same conclusion. Consider the conditions that b_2 must satisfy simultaneously:

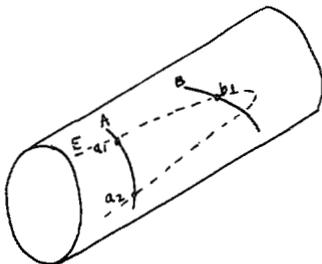


Figure 8. Alternative construction for proving the parallel-line observation.

- Obviously, b_2 has to be on the B scan.
- If the segment $\overline{a_2b_2}$ is to be parallel to $\overline{a_1b_1}$ then it must be in the plane P defined by the three points a_1, b_1, a_2 . Therefore, b_2 is constrained to lie on the ellipse E formed by the intersection of the plane P and the cylinder.
- Therefore, b_2 must be one of the two possible intersections of the B ellipse and the E ellipse. But, b_1 is already (by construction) one of these intersections, so b_2 must be the other *and* it must be on the line parallel to $\overline{a_1b_1}$ going through a_2 .

Clearly, b_2 cannot generally be chosen to satisfy all those conditions.

The, perhaps unsurprising, conclusion is that there is a good reason why the implemented method works well.

2.4 Getting the bends

We have used the method described in section 2 to segment a tube made up of straight and curved sections. The desired result is a list of the scans on the straight sections. We detect the onset of a bend by examining the angles that the computed axes make to the global coordinate axes. The axes on the straight sections will have nearly constant values of these angles while the axes computed along the bends

will have rapidly changing values. Figure 9 shows the axis angles computed from the examples in figure 5 and the segmentation into linear segments. The angles are relatively noisy due to the inaccuracies in the scan data, but during the straight segments the deviation from the average angle is within 2 or 3 degrees. We use a simple split and merge algorithm to find segments of nearly constant angle. We have used this method in the results in figure 5 with quite satisfactory performance.

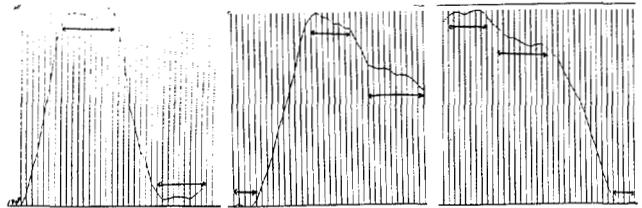


Figure 9. Axis angles computed for the examples of Figure 5. The straight segments are indicated in each graph, and correspond to the cylinder segments found in Figure 5.

3. Fitting, and why it is not

Initially, we attempted to locate the cylinders using ellipse fitting. Bob Bolles of SRI has reported satisfactory performance with this method. A description of his approach can be found in [Bolles 86]. We attempted to replicate his approach and found that while it was possible to detect cylinders that way, the method was error prone and inaccurate. It worked for Bolles because of the high quality of the data he was using (obtained from a Technical Arts scanner). With a much less accurately calibrated scanner at MIT, only about half the scans could be successfully fitted and the expected error in the axis direction was around 20 degrees. In this section, we describe some of these attempts.

First we attempted to ignore efficiency and applied the two best known algorithms for fitting conics [Bookstein 79] and [Sampson 82]. The results were dismal. The methods have slow convergence and they are very sensitive to error. They worked quite well with idealized sampled data, but they amplified noise by an order of magnitude. The main reason seems to be that realistic scan data only covers a small portion of the outline of the ellipse. These methods work fairly well when data covering most of the perimeter is available.

At this point, we attempted a new approach based on using only four points of data and exploiting the known radius. The hope was that this would be much more efficient and not noticeably less accurate. The basic idea stems from the observation that the perpendicular distance of any

point on the cylinder from the axis line is R , the cylinder radius. You need four points to constrain the line because the line has four degrees of freedom. The points can be drawn from anywhere on the cylinder. Let t is a unit vector along the cylinder axis, q is a point on the cylinder axis chosen so that its position vector is perpendicular to t , and p_i , ($1 \leq i \leq 4$) are the points on the cylinders. We want to find t and q that satisfy the following six constraints:

$$\begin{aligned} \|(p_i - q) \times t\| &= R \\ q \cdot t &= 0 \\ \|t\| &= 1 \end{aligned}$$

The first four of these constraints state that the perpendicular distance from the p_i to the axis line is R , the cylinder radius.

As expected, this method was more efficient but it proved sensitive to the choice of points on the scan. Again, given points distributed around the perimeter improved its performance.

We tried several other variations of these methods, all with similar (non)performance. Ultimately, we think that methods that depend on obtaining accurate conic fits and operating on the conic parameters are doomed to failure.

Before discovering the method described in section 2, the most effective technique we found involved connecting the endpoints of the scans and locating sets of them that are parallel. If the scans of a tube are all taken while the scanner is translating, this method provides a very quick way of segmenting the data. We later found out that this is precisely the method used by Agin in his early Stanford thesis [Agin 72].

4. Surface Rulings

Finally, we observe that our technique for finding cylinder axes from data essentially reduces to finding rulings on the cylinder surface from sparse scans, and using the simple relationship between the surface rulings and the axis of the cylinder to deduce the position and orientation of that axis. This observation suggests that more complex surfaces could also be analyzed in this manner. For example, cones could be deduced by finding rulings on the surface that intersect at a common point. It may also be possible to analyze generalized cylinders in this manner, and we are currently investigating this extension.

Bibliography

[Agin 72] Gerald J. Agin, "Representation and Description of Curved Objects", Stanford A I Memo 173, October 1972.

[Bolles 86] Robert C. Bolles, "Three-Dimensional Locating of Industrial Parts", in *Robot Sensors: Vol 1, Vision*, edited by Aln Pugh, IFS Publications Ltd and Springer Verlag, 1986.

[Bookstein 79] Fred L. Bookstein, "Fitting Conic Sections to Scattered Data", *Computer Graphics and Image Processing*, Vol 9, pp 56-71, 1979.

[Sampson 82] Paul D. Sampson, "Fitting Conic Sections to "Very Scattered" Data: An Iterative Refinement of the Bookstein Algorithm", *Computer Graphics and Image Processing*, Vol 18, pp 97-108, 1982.