An Automatic Tube Inspection System That Finds Cylinders in Range Data[1]

W.E.L. Grimson
MIT AI Lab
Cambridge MA

T. Lozano-Pérez
MIT AI Lab
Cambridge MA

N. Noble
SOCS Research
Los Gatos CA

S.J. White
TASC
Reading MA

Abstract

We describe a system that automatically inspects pieces of formed tubing. To locate the part, the system automatically identifies cylindrical tube sections from sparse range samples, taken with an active scanning device, and uses this to roughly locate the part under the sensor. This localization is used to determine a detailed scanning strategy, which is executed to acquire more precise and detailed information. This data is then used to verify specifications on components.

1 The Problem

Tubing for an aircraft are formed by inserting a straight piece of pipe stock, of some constant radius, into an automatic tube bending machine. This machine then executes a numerical control algorithm, that advances a portion of the through a bending device, bends it by some predetermined amount to create a toroidal section, then iteratively rotates, advances and bends the pipe. The specification of the feed length, bend rotation and bend angle is called an LRA (Length, Rotate, Angle) model. It can be used to specify any tube form which can be manufactured using the bending machines.

Prior to this work, completed tubes were inspected by hand: a human operator would place the pipe on a flat table over a master blueprint, align the pipe by eye, then visually check that the pipe agreed with the blueprint. Since the tubes can bend in 3D, this method may need several alignments and may still be inaccurate on portions of the tube that cannot be placed flat on the support surface under any orientation of the tube.

Our goal was to automate the inspection process, subject to the following specifications:

- Accuracy of .004 inches on the linear segments.
- Speed of 1 minute per part.
- Arbitrary part placement and pose.
- No part "programming" , only the part specification or bending program is available.

The input to the system is:

- A tube placed in any stable position within a 2.5' by 4' by 14' volume,
- an object model, i.e. a blueprint or physical model of the ideal tube, in this case an LRA model,
- specifications on the components of an ideal object, e.g. tube radius, length of straight tube segments, angle and radius of bend of toroidal connections and relative orientation of straight segments.

The system's output is simply another LRA model, reflecting the actual geometry of the measured tube. This, when compared to the specification LRA model, can be used for quality control, or to adjust the bending program to bring the process closer to specification.

2 The Approach

Our approach consists of a series of stages:

Coarse Sensor Input: We use an active laser scanning device as our sensory input. In the first stage, the scanner provides a sparse set of measurements from the tube, taken at a large standoff from the part.

Cylinder Extraction: From the coarse data we hypothesize locations and orientations of cylindrical tube segments, measured in world coordinates.

Matching: These cylinder hypotheses are matched against our known model, to give an estimate of the position and orientation of the complete tube.

Scan Planning: We next plan a scanner path that will trace along the tube, while keeping the scanner aligned with the current orientation of the tube section, and avoiding collisions with the tube.

Fine Sensor Input: By tracking the scanner at close range along the tube, we obtain high accuracy information about the location of the tube axis. This can be used to generate a tube model.

Inspection: This model can be matched to the specification LRA model to insure no deviation exceeds specified limits. Detected deviations can also be used to adjust the bending program so that subsequent parts more closely match the specified LRA model.

We now describe each of the components in turn.

[1] This report describes research done in part at the AI Lab of MIT. Support for the laboratory's research is provided in part by ONR contract N00014-91-J-4038.

1063-6919/93 $03.00 © 1993 IEEE

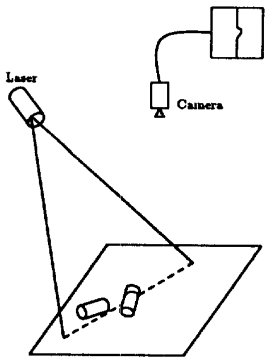

Figure 1: Schematic of laser sensing system. For each position of the plane of light, a scan of points in the camera can be processed to determine three dimensional position.

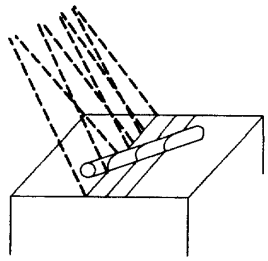

Figure 2: Schematic of sensing using four stand off scanners. The lasers are aligned so that they create a single plane of light, which is sampled at .08 inch spacings.

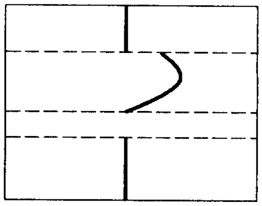

Figure 3: Example of a laser slice. A representative image for a cylindrical object under the scanner is shown in dark. A jump in distance greater than 0.5 inches in height (the top dotted line) or a gap in the data greater than 0.25 inches (the bottom two dotted lines) signifies an end to a data segment.

2.1 The Laser Scanner

Our input device is a structured-light laser scanner manufactured by Technical Arts Corporation. It operates by scanning a laser beam using an oscillating mirror such that a plane of light is produced. A video camera is placed at an angle to this plane, with part of the plane in the camera field of view. When an object is placed in this region, points in the camera image illuminated by the laser unambiguously correspond to fixed 3D scene points (Fig. 1). Correspondences between scene pointsand image points are calculable by using a nonlinear projective transform, which can be determined by scanning an object of known form. Since the deflection of the beam in the image is in a known direction, one can process many such points in a single scan. In this case, the device produces 240 3D measurements for any single scan.

When the scanner is transported, any point in a 3D volume can be measured by adding the scanner data to the position information of the transport. When the scanner is not moving, we call the scan data **static**. In this case, the scan data corresponds to a planar cross-section of the scene. When the scanning takes place while the transport is moving, the scan is called **dynamic** and the cross-section information is more complex. In dynamic scans, the sweep profile of the oscillating laser mirror determines the scan profile.

2.2 The Coarse Scan

To obtain dense, accurate information about the geometry of the tube, we need the scanner to be close to the tube, with the plane of the laser perpendicular to the axis of the tube, as this provides the most reliable information about the tube shape. Because the tubes can bend in 3D, and can be placed in arbitrary position on the support table, within a $2.5 \times 4 \times 14'$ volume, we cannot do this without fear of hitting the tube. Hence, in a first pass, we transport four structured-light sensors over the work surface, at a large standoff from the surface, so that we are guaranteed not to hit the tube, for any orientation. For efficiency, we scan the entire work volume at regular intervals, sampling planar cross-sections of the work surface above table height at a 0.08 inch spacing (Fig. 2). Thus at each position, we obtain a set of pixels in the cameras corresponding to the intersection of the laser with the scene, whose position encodes the height of the intersected surface.

Next, we segment the illuminated image pixels for each intersection of the laser plane and the scene, by treating the data from each position of the system as a set of piece-wise continuous segments, and finding discontinuities in the set of pixels. In particular, we look for breaks in the image segments that correspond to scene discontinuities at least 0.5 inch in height or 0.25 inch in width. These correspond either to sudden jumps in the scene itself or to occlusions of the laser corresponding to such jumps. Once we have segmented the data into contiguous sections, we can remove segments that lie entirely on the support plane. The data within each remaining scan segment are assumed to be from a single cylindrical cross-section of the tube (Fig. 3). The sparse scanners are arrayed with their scan planes aligned along a line perpendicular to the direction of travel. The fields of view of adjacent scanners are partially overlapped, but the scans are acquired sequentially over time. The output from this stage is a set of data segments from the surface of the tube (Fig. 4).

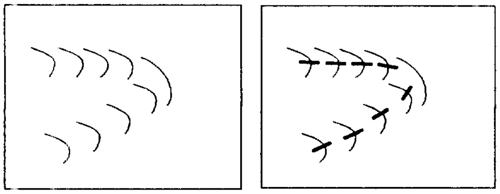

Figure 4: Left: Example of data segments from the tube surface. Right: Hypothesized axis directions.

2.3 Finding the axes

To locate the tube, we need to find the positions and orientations of the cylinder axes of each of the straight legs, so that we can match these to a model description and compute the transformation mapping model to scene. By segmenting the scans, we obtain connected sets of points that come from the surface of the tubing. The trick is to separate out data due to the straight legs from data due to the toroidal bends, and then determine the parameters of the cylindrical sections.

We focus on locating the cylindrical sections from the range data. We do this 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. A common approach to constructing a cylinder axis from one or more scans is to fit an ellipse to the scan. The parameters of the ellipse can be used to solve for the cylinder axis [1, 2, 9]. 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. However, this technique is limited to static scans and is subject to numerical accuracy problems, especially when only small segments of the ellipse are available.

In [8], we developed a simple technique for finding the orientation of an axis given two or three scans on the cylinder. The technique is very stable numerically. Additionally, the technique works not only with static scans, but also with a broad class of dynamic scans.

2.3.1 Characterizing the cylinder axis

The method we use for this task is based on the following geometric question (refer to Fig. 5):

Take two scans (A and B) on a cylinder, obtained as the intersection of two different illumination surfaces 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 $\overline{a_1b_1}$ and $\overline{a_2b_2}$ parallel iff they are also parallel to the cylinder axis?

To answer this question, it suffices to consider a cylinder of unit radius, centered at the origin, with axis along the $\hat{z}$ axis. Points on the cylinder are represented by $(cos\theta_i, \sin\theta_i, h_i)$. We are free to choose a coordinate system with the $\hat{x}$ axis such that $\theta_2 = -\theta_1$, hence

$$\begin{aligned} \mathbf{a}_1 - \mathbf{b}_1 &= (0, 2\sin\theta_1, h_1 - h_2) \\ \mathbf{a}_2 - \mathbf{b}_2 &= (\cos\theta_3 - \cos\theta_4, \sin\theta_3 - \sin\theta_4, h_3 - h_4). \end{aligned}$$

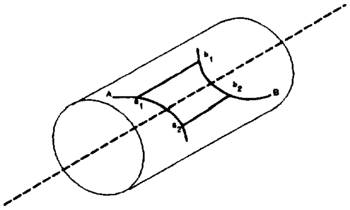

Figure 5: Two scans A and B on a cylinder. The question is whether the line segments a_1b_1 and a_2b_2 are parallel only when they are also parallel to the cylinder axis.

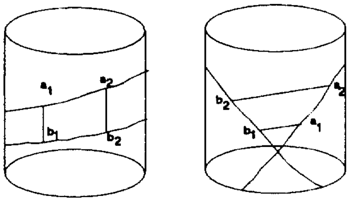

Figure 6: Two solutions to the problem of parallel lines, whose endpoints lie on the surface of the cylinder. In the case on the left, the lines are also parallel to the cylinder axis. In the case on the right, they are not.

These vectors are parallel if and only if:

$$\begin{aligned} 2\sin\theta_1(\cos\theta_3 - \cos\theta_4) &= 0 && (1) \\ (\cos\theta_3 - \cos\theta_4)(h_1 - h_2) &= 0 && (2) \\ (\sin\theta_3 - \sin\theta_4)(h_1 - h_2) &= 2\sin\theta_1(h_3 - h_4) && (3) \end{aligned}$$

We can exhaustively examine the conditions needed for parallelism:

Case 1: If $\theta_4 = \theta_3$ then $\mathbf{a}_2 - \mathbf{b}_2$ is parallel to $\hat{\mathbf{z}}$, as desired.

Case 2: If $h_1 = h_2$ and $\sin\theta_1 = 0$, then $\mathbf{a}_1 = \mathbf{b}_1$.

Case 3: If $\theta_4 = -\theta_3$, then

$$\begin{aligned} \mathbf{a}_1 - \mathbf{b}_1 &= (0, 2\sin\theta_1, h_1 - h_2) \\ \mathbf{a}_2 - \mathbf{b}_2 &= (0, 2\sin\theta_3, h_3 - h_4). \end{aligned}$$

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

$$\sin\theta_1(h_3 - h_4) = \sin\theta_3(h_1 - h_2)$$

we have a valid set of points, whose separation vectors are parallel to each other, but not to the cylinder axis.

Fig. 6 shows two different classes of solutions. We need an additional condition that will allow us to distinguish the case on the right from that on the left.

2.3.2 Surface Rulings

In the case on the right of Fig. 6, the two lines will only intersect the cylinder at the chosen points. Elsewhere, the lines will either lie inside or outside of the

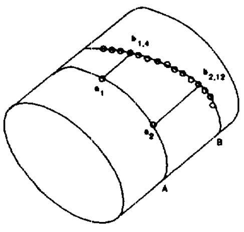

Figure 7: A linear time algorithm for finding the axis.

cylinder itself. In the case on the left, the entire lines lie along the surface of the cylinder. Thus, suppose we take a third scan of the cylinder, and further suppose that on that scan, we can find a point $\mathbf{p}_5$ such that $\mathbf{a}_1 - \mathbf{p}_5$ is parallel to $\mathbf{a}_1 - \mathbf{b}_1$. Then we have

$$\begin{aligned} \mathbf{a}_1 - \mathbf{p}_5 &= (\cos\theta_1 - \cos\theta_5, \sin\theta_1 - \sin\theta_5, h_1 - h_5) \\ \mathbf{a}_1 - \mathbf{b}_1 &= (0, 2\sin\theta_1, h_1 - h_2) \end{aligned}$$

Thus, these two lines being parallel implies $\theta_1 = \pm\theta_5$. If $\theta_1 = \theta_5$, then clearly $\mathbf{a}_1 - \mathbf{p}_5$ and $\mathbf{a}_1 - \mathbf{b}_1$ are both parallel to $\hat{\mathbf{z}}$ and we are done. If $\theta_1 = -\theta_5$, then

$$\mathbf{a}_1 - \mathbf{p}_5 = (0, 2\sin\theta_1, h_1 - h_5).$$

Now, $\mathbf{a}_1 - \mathbf{p}_5$ and $\mathbf{a}_1 - \mathbf{b}_1$ being parallel implies either $\sin\theta_1 = 0$ or $h_1 - h_5 = h_1 - h_2$. But if $\mathbf{a}_1 \neq \mathbf{p}_5$, 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 on a cylinder, obtained as the intersection of three different surfaces of illumination 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 line segments are also parallel to the cylinder axis.*

We observe that this technique for finding cylinder axes from data essentially reduces to finding rulings on the cylinder surface from sparse scans. Given such rulings, we can use 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.

2.3.3 Solving for the cylinder axis

Our method looks for surface rulings on the cylinder, since these are parallel to the cylinder axis.

Step 1: Consider Fig. 7. Given two scans, pick two points on one scan, $a_i (i = 1, 2)$; the points should be widely separated on the scan, but lie on the common overlap of the two scans (i.e. the intersection of the projection of each scan onto the cylinder base). We must choose $b_i (i = 1, 2)$ on the B scan such that the lines $\overline{a_1 b_1}$ and $\overline{a_2 b_2}$ are parallel. Let $b_{i,k} (k \leq n)$ indicate the choice of the k^{th} point on the B scan as b_i. Then, for each point $b_{i,k} (k \leq n)$, construct the unit vector $v_{i,k}$ from a_i pointing at $b_{i,k}$. 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. We start with $k = 0, l = 0$, that is, the initial B points are $b_{1,0}$ and $b_{2,0}$. The basic loop of the algorithm increments k and 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.

Finding the parallel surface lines

```
$l_{best} = 0$
$dot_{best} = v_{1,0} \cdot v_{2,0}$
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($v_{1,k-1}$ and $v_{2,l_{best}}$)
End Loop
```

The value of dot_{best} should approach 1.0 monotonically and then drop off. At the peak, both the $v_{i,k}$ are estimates of the cylinder axis direction. This estimate can be refined to sub-sample resolution by interpolating between the points $b_{i,k}$. 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 little rotation of the surface of illumination between the scans, then the initial choice of k and l to be greater than 0 can significantly reduce the computation.

Since 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. Noise can also affect the accuracy of the computed rulings. Since we are mainly interested in the direction of the axis, which should be parallel to the rulings, we can reduce the effects of error by averaging the results obtained from a variety of choices of the a_i.

Step 2: Once we find parallel lines, we can hypothesize that these are actually rulings on the surface of the cylinder. Next we must verify that this is actually true. In [8] we show that this can be done by intersecting the infinite line through either of the hypothesized rules with a third scan. If either extension does intersect the scan, then we can deduce that we have a pair of rulings that are parallel to the axis of the cylinder. If neither extension intersects the third scan, we may still be able to use the hypothesized rulings, but we cannot guarantee that they are in fact rulings. Furthermore, if the scans are nearly parallel (e.g. with a straightforward translation of the sensor configuration), there is no possibility that the scans can cross on a single leg.

In this case, two scans are sufficient to uniquely determine the axis orientation, i.e. we cannot possible have the situation shown in the right part of Fig. 6.

Step 3: Given parallel rulings, we know that the axis direction is the same as the rulings. Now we can cluster axes hypotheses from successive pairs of scans (Fig. 4). If $\hat{\mathbf{h}}_i$ denotes a unit vector in the direction of the cylinder axis for the i^{th} scan, then we are seeking consecutive scans $\hat{\mathbf{h}}_i, \hat{\mathbf{h}}_{i+1}, \ldots \hat{\mathbf{h}}_k$ such that $\hat{\mathbf{h}}_j \cdot \hat{\mathbf{h}}_\ell \geq \alpha, \forall i \leq j < \ell \leq k$, where α is some threshold. We use the direction, since we don't know the offset of the axis.

Each cluster of consecutive scans giving rise to a similar axis constitutes a different hypothesized cylinder segment. If a sequence of scans pairwise give rise to roughly the same axis direction, then the likelihood of pairs of line segments being parallel to one another, but not parallel to the axis cylinder becomes vanishingly small, and it is possible to omit step 2 above.

Step 4: 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. In other words, points on the actual axis are given in sensor coordinates as $\mathbf{a}(\alpha) = \mathbf{p} + \alpha\hat{\mathbf{t}}$ where $\hat{\mathbf{t}}$ is a unit vector in the direction of the axis and $\mathbf{p}$ is a displacement vector from the origin of the coordinate system. The method in the previous steps allows us to determine $\hat{\mathbf{t}}$. Any of several methods will serve to determine $\mathbf{p}$. In our testing, we have simply used the position of the average point of the scan as a (very) rough estimate of the center of the cylinder. Alternatively, we could use a straightforward Hough transform technique to determine the offset of the cylinder axis.

2.4 Surface Rulings

Note that this technique for finding cylinder axes from data essentially reduces to finding rulings on the cylinder surface from sparse scans. If we can find such rulings, we can use 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.

2.5 Getting the bends

We have used the described method to segment a tube made up of cylindrical and toroidal sections. The desired result is a list of the scans on the straight (or cylindrical) sections. We detect the onset of a bend by examining the angles that the computed axes make with the global coordinate axes. The axes on the straight sections will have nearly constant values of these angles while the axes computed along the toroidal bends will have rapidly changing values. Fig. 9 shows the axis angles computed from the examples in Fig. 8 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 Fig. 8 with quite satisfactory performance. The output from this stage is a set of cylinder axes, including an axis direction, an offset from the origin, and the positions of the endpoints of the axis.

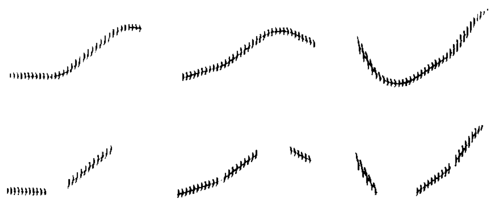

Figure 8: Examples of tube axes computed from real data.

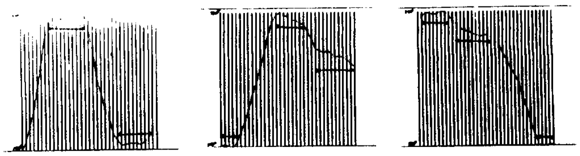

Figure 9: Axis angles computed for the examples of Figure 8. The straight segments are indicated in each graph, and correspond to the cylinder segments found in Fig. 8.

3 Matching the Model

Given a set of measured cylinder axes, we can now match these against a part model to determine its pose. We represent a model by a set of axes, given by their endpoints: $\{(\mathbf{b}_i, \mathbf{e}_i)\}, i = 1, \ldots, n$ with an associated unit tangent $\hat{\mathbf{t}}_i$ in the direction of $\mathbf{e}_i - \mathbf{b}_i$. We can similarly represent our data as $\{(\mathbf{B}_j, \mathbf{E}_j)\}, j = 1, \ldots, m$. Our goal is to find a correspondence assigning each data feature a matching model feature

$$f : \{1, \ldots, m\} \mapsto \{1, \ldots, n, *\}$$

(where $*$ indicates that the data fragment is spurious), such that there is a rigid transformation $\mathcal{T}$ such that the transformed model leg lies within ϵ of its matching data leg, and that the projection of the data leg onto the model leg lies (modulo error) along the actual leg. Note that in searching for such a correspondence, we may have to consider data legs in which we reverse the roles of $\mathbf{B}$ and $\mathbf{E}$, since the endpoints are not distinguishable.

There are many methods for finding correspondences between data and model features subject to such constraints, including interpretation tree search, or use pose space search (see [3] for discussions of these and other approaches). Here we use a simple variation on what is now commonly termed an alignment search.

We simply search over all pairs of correspondences between data and model cylinder axes. For each, we check that the pairings meet some simple geometric constraints, which serve to remove geometrically unviable solutions. For each remaining pair of correspondences, we find the transformation that brings the matched model cylinders into alignment with their corresponding data cylinders, and then check that the remainder of the aligned model is consistent with the data.

3.1 Constraints Used in Matching

To match pairs of model and data axes, we use a series of simple constraints. First, we require that a candidate matching of a sensory and model leg have lengths that match within some error bounds, i.e. if

$$.1\|\mathbf{b}_j - \mathbf{e}_j\| \leq \|\mathbf{B}_{f(j)} - \mathbf{E}_{f(j)}\| \leq 2\|\mathbf{b}_j - \mathbf{e}_j\|.$$

A pair of model axes can match a pair of data axes if the angle between each pair is roughly the same. We implement this using the dot product of the unit vectors in the direction of the axes, i.e.

$$\|(\mathbf{T}_i \cdot \mathbf{T}_j) - ((\mathbf{t}_{f(i)}) \cdot (\mathbf{t}_{f(j)}))\| \leq \epsilon_a.$$

We can also use the distance between legs as a constraint, which we incorporate based on the shortest chord joining the two leg axes. Fig. 10(a) shows this chord XY, referred to as the "base" of the leg pair. It is orthogonal to both leg axes. The points X and Y need not lie between the endpoints of the feature legs (AB or CD). There are four distances saved as feature distance constraints. These are the minimum and maximum distances between the intersection of the base and the legs (X and Y) and the endpoints. These are shown in the figure as the distances a, b, c, and d. For each such measurement, we invoke a constraint which requires that the measured distance for a pair of data legs be in rough agreement with the corresponding distance for the associated pair of model legs.

Another binary constraint used in the match pruning is the base length. In the figure, this is the distance between X and Y. All the sensory binary length constraints are compared to the candidate model pair values with error bounds built into the model data set.

Legs pairs in the model or sensory sets which are nearly parallel are removed due to the instability of any transformation that would result by using them. When the leg lengths are nearly equal, there are two possible transforms to align the model and sensory leg pairs. When the legs are also nearly orthogonal, an additional matching ambiguity exists, raising the number of possible transforms to four. When such conditions arise, each possible transform is tested.

3.2 Model Transformation and Testing

Combinations of legs which survive the constraint pruning are used to create transforms of the model to the sensory data. These transformations carry the "base point" — the midpoint of the base — of the model legs to the base point of the sensory legs. This is shown in Fig. 10(b) as point O. The transforms also rotate the base of the model legs into alignment with the base of sensory legs (indicted as the X axis in the figure), and they carry a bisector of the model legs to one of the four bisectors of the sensory legs. A bisector, shown as the Y axis in the figure, is a ray emanating from the base point which bisects one of the angles formed by the projections, along the base and onto a plane normal to the base, of the axes of the legs.

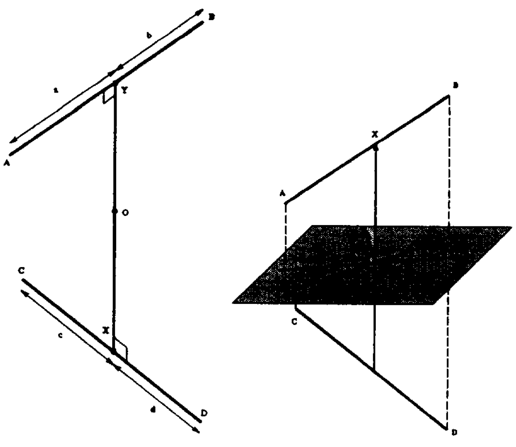

Figure 10: (a) Distance Constraints, (b) Transform Basis

If such a transformation brings any leg substantially below the table surface or otherwise outside the work volume, the transform is rejected. Once the entire model data set is transformed, each corresponding model and sensory leg is tested for good alignment. Each model/sensory pair is tested and assigned a numeric measure of fit, based on the angle between the two leg axes as well as the distance between the legs, determined in much the same way as the base distance was calculated in the constraint steps described above. If any pair mismatch exceeds a threshold, the transform is rejected. Of the accepted transforms, the one with the best match measure (based on the base distance matching and a measure of the total model leg length matched) is chosen as the solution.

The purpose of matching is to determine the alignment of the tube on the inspection table. Although the transformed model would normally suit this purpose, the physical tube often is incorrectly formed, or distorted by gravity or friction, such that the best match is still measurably different than the model. To correct for this, a final step transforms each model leg to bring it into as close a fit as possible to the measured data. This completes the model location processing, the output of which is a complete transformed model which is very close to the form and orientation of the tube placed arbitrarily on the table.

4 Scan Planning

In order to measure the tube form, a large amount of data is needed from the cylindrical legs. The toroidal bend sections are not used in the measurement, since the bending process tends to deform the tube cross-section. Thus, it is important to get as much data as possible on the cylindrical sections to insure accurate results. Since the accuracy of the scan data is inversely proportional to the measurement range of the detail sensor and its standoff from the part, the scanner must be transported over each leg of the part in close proximity to the leg. The laser plane orientation must also be roughly aligned to the tube axis, since this will produce data segments with the least amount of distortion and hence will lead to more accurate estimates of tube axes. This requirement means that an inspection plan must be generated from the localized model information in order to move a 4-axis (three Cartesian and z-axis rotation) gantry over the tube sections. While the scanner is being transported, scan data is acquired dynamically.

The first pass produces a model of the location and orientations of all of the tube sections. From this information we want to generate a path plan for the detail scanner. We considered several approaches here.

The first approach used configuration-space methods to plan a series of translations and rotations of the gantry such that the scanner would track along the tube, while guaranteeing (when possible) that the tube would be visible both to the camera and to the laser, and such that projection of the normal to the plane of the laser onto the image plane would align with the projection of the axis of the tube cylinder onto the same plane While we were able to construct and test such a planner, experimentation on real tubes suggested that simpler methods were generally possible.

Hence, the solution finally adopted was to simply pick as the scanner path, the actual positions of the localized tube, with a translation of 18" (the standoff of the scanner) in the Z direction. The scanner orientation is simply the projection of the tube model leg axis projected on the X-Y plane. This proved to be generally sufficient, although there are possible convoluted tube shapes for which such a scheme would not be able to acquire data along all cylindrical sections.

Once the path plan is generated, it is tested against the localized model and the known size of the scan head to insure collision-free scanning.

5 The Second Scan

Once the inspection plan is complete, the detailed scan data is acquired. Since the accuracy of the axis measurements depends, in part, on getting a large amount of data covering as much of the straight cylindrical tube sections as possible, about 200 scan points are acquired for each scan cross-section. A scan is taken about every 0.1" along the length of the tube leg. Therefore, on a 6" leg, 120,000 individual 3D data points are acquired.

Cylinders are fit to the data, using least squares methods, and a detailed scanner sensory model produced from these data. This model is the end-product of the inspection process. It is used both to determine the acceptability of the part and to produce a "corrected" bending program for subsequent part runs.

6 Testing and Conclusions

We have tested this method as follows:

Coverage. The scanner correctly inspected more than 400 different designs.

Speed. The ATS inspected a typical 6' long, 5 bend tube in under 2 minutes (total sparse/detail scan and data processing time).

Repeatablity. When a tube is inspected multiple times, the data vary by less than 0.003" (3 sigma).

Accuracy. The system was accurate on tube axis measurements to ±0.008" (3 sigma)

Sizes. Tube diameters varied between $\frac{3}{16}$" and 3".

Bends. Models with up to 13 bends.

Overall Tube Length. Up to 13'.

Our conclusion is that the system accurately and reliably inspects a wide range of tubes, demonstrating the utility of the approach. The method can be extended to handle dynamic scans, and more general shapes, and is the subject of a forthcoming paper.

References

[1] Bolles, R. C., "Three-Dimensional Locating of Industrial Parts", in *Robot Sensors: Vol 1, Vision*, Alan Pugh (ed), IFS Publications Ltd. and Springer Verlag, 1986.

[2] Bookstein, F. L., "Fitting Conic Sections to Scattered Data", *CGIP*, Vol 9, pp 56–71, 1979.

[3] Grimson, W.E.L., 1990, *Object Recognition by Computer*, MIT Press, Cambridge.

[4] Grimson, W. E. L. and T. Lozano-Pérez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data", *IJRR*, **3**, No. 3, 3 – 35, 1984.

[5] Grimson, W. E. L. and T. Lozano-Pérez, "Localizing Overlapping Parts by Searching the Interpretation Tree", *IEEE PAMI*, **9**, No. 4, 469–482, 1987.

[6] Huttenlocher, D.P. and S. Ullman, 1987, "Object Recognition Using Alignment", *Proc. 1st ICCV*, pp. 102-111.

[7] Huttenlocher, D.P. & S. Ullman, 1990, "Recognizing Solid Objects by Alignment with an Image," *IJCV* **5**(2):195–212.

[8] Lozano-Pérez, T., W. E. L. Grimson, and S. J. White, "Finding Cylinders in Range Data", IEEE Int. Conf. Rob. Autom., Raleigh, NC, April 1987, pp. 202–207.

[9] Sampson, P. D., "Fitting Conic Sections to "Very Scattered" Data: An Iterative Refinement of the Bookstein Algorithm", *CGIP*, Vol 18, pp 97–108, 1982.