On Seeing Spaghetti:
A Novel Self-Adjusting Seven Parameter Hough Space for Analyzing Flexible Extruded Objects

John R. Render
Department of Computer Science
Columbia University
New York, NY 10027
kender@cs.columbia.edu

Rick Kjeldsen
Exploratory Computer Vision Group
IBM Thomas J. Watson Research Center
P.O. Box 704 Yorktown Heights, NY 10598
femk@watson.ibm.com

Abstract

We present a model for flexible extruded objects, such as wires, tubes, or grommets, and demonstrate a novel, self-adjusting seven-dimensional Hough transform that derives and analyzes their three-space curved axes from position and surface normal information. The method is purely local and is very cheap to compute. The model considers such objects as piecewise toroidal, and decomposes the seven parameters of a torus into three nested subspaces, the structure of which counteract the errors implicit in the analysis of objects of great size and/or small curvature. We believe it is the first example of a parameter space structure designed to cluster ill-conditioned hypotheses together so that they can be easily detected and ignored. This work complements existing shape-from-contour approaches for analyzing tori: it uses no edge information, and it does not require the solution of high-degree non-linear equations by iterative techniques. Most of the results, including the conditions for the existence of more that one solution (phantom "anti-tori"), have been verified using a symbolic mathematical analysis system. We present, in the environment of the IBM ConVEx system, robust results on both synthetic CAD-CAM range data (the hasp of a lock), and actual range data (a knotted piece of coaxial cable), and discuss several system tuning issues.

1. Introduction

We consider the problem of analyzing dense depth images to determine the parameters of flexible extruded objects. Our approach views such objects as being piecewise toroidal, since the torus is the simplest solid geometric object whose generating axis exhibits curvature. This is justified by our observation that many piecewise sections of flexible extruded objects have co-planar spines, either because they are easy to manufacture that way, or because they represent a minimum energy configuration (these reasons are often equivalent). However, a torus is an object with seven free parameters: three of position, two of orientation, one of size, and one of relative "thickness". This pushes the limits of the current state of the art in parsing high-degree parametric object models, particularly since the parameters of position and size are potentially unbounded, especially for objects that are nearly cylindrical.

Following what might be called the principle of least variability, we first recover the thickness parameter, since among all seven it is the one parameter most likely to be constant across the toroidal pieces. The method exploits a novel but efficient algorithmic interpretation of a result in differential geometry called Meusnier's theorem. Then, using the knowledge gained from deriving the thickness, we recover the next most well-behaved parameters, the orientation and size. These are computed simultaneously and in a manner that compensates for the ill-conditioning of orientation estimates when size is large. Lastly, using knowledge of both thickness and size, we recover the position of the local toroidal section, again automatically compensating for the ill-conditioning of large and/or nearly straight objects.

The significance of the work rests on its two principal results: the elegance and speed of the thickness-finding transform (reviewed only briefly here, for more details see [Kender and Kjeldsen, 1990], and the novel way in which the parameter space structure decomposes a difficult seven-dimensional problem so that ill-conditioned hypotheses cluster together for easy detection and removal. The work as a whole is applicable in vision systems wherever depth and surface orientation (however sparse) are obtainable, particularly for those cases where object boundaries are occluded, and where contour-based methods therefore fail.

1.1. The Torus in Brief

We adopt the terminology of DoCarmo [1976], and review the results on tori that we will exploit.

A torus is a solid of revolution formed by a generating circle ("minor circle") of radius *r* being swept in a circle of revolution ("major circle") of radius *a* (Note that we have reversed the sense of/' & *a* from DoCarmo). Generally speaking, we will assume that $r<a$, that is, that the "donut" has a "hole". This constraint will be exploited in the decomposition of the parameter spaces.

Surface properties are best represented in the framework of the following parameterization, which identifies the center of the torus (and hence, the center of the major circle) with the

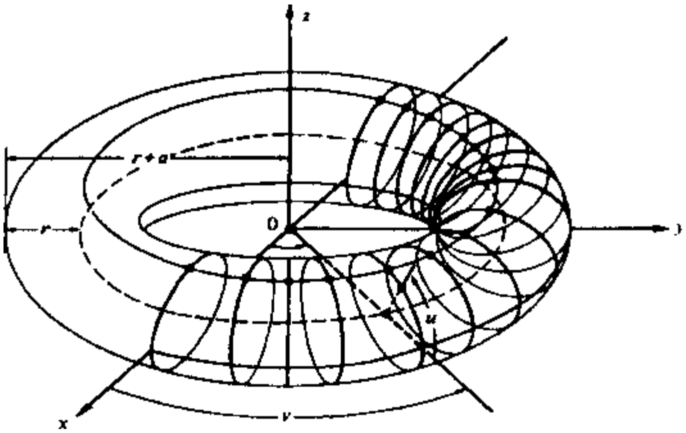

Fig. 1: Torus terminology.

center of a cylindrical coordinate system, making *a* to be cylindrical radius and v as cylindrical angle; the axis of the torus becomes the remaining coordinate *z.* We parameterize the surface of the minor circle by the angle *u;* unlike the angle v, its origin is critical, and is set to equal 0 when it is most distant from the torus center. The locus of points with a given v lie on a "meridian". The locus of points with a given *u* lie on a "parallel".

As with any other regular surface, each point of the torus has two principal curvatures, with directions of principal curvature that are perpendicular to each other. Every point has one positive curvature, k_1, which is oriented along the meridian. The other curvature of the point, k_2, is more elaborate, although it is always oriented along the parallel, k_2 is 0 at the "top" and "bottom" parallels of the torus, where the surface is locally cylindrical. It attains its extreme positive value at the "outermost" parallel, and its extreme negative value at the "innermost" parallel. We note and will heavily exploit the observation that the largest positive curvature at any point is always the value of *k\.*

We will adopt the following notation. A point on the torus is denoted as *Pi;* it is considered to be vector in three space. We will abbreviate the vector P_i-P_j as P_{ij}. Likewise, we will refer to the unit normal vector of the surface at *Pi* as *Ni,* and a similar comment applies to vectors of the f*orm Ni.j.* Note that the values of the points and the normals are available as input data, measured in the coordinate system of the imaging apparatus. We will often refer to the "translated" points *Ti = Pi - rNi*; if the *r* is the true minor radius, then *Ti = Si,* that is a point on the torus spine.

2. Computing Minor Radius

It is not hard to show that when computing the parameters of a two-dimensional surface in three space, the information that a point is on the surface can be used to fix one parameter, and the information that a given vector is the normal to the surface at a point can be used to fix two parameters. (Analogous statements can be made about curvature, but we avoid curvature information because of its intrinsic susceptibility to noise.) Thus, a sphere is uniquely determined by four points, or, more simply, by two points and the normal at one of them. Thus, two points and their normals are insufficient to uniquely determine a torus, since they fix only six parameters. However, three points and their normals in general overdetermine the torus in the following way. There must exist a local space circle that serves as the torus spine, and it must be equidistant from all three points (this common distance is the value of r). But, in addition, this space circle must have the property that at the point of closest approach to each of the *Pi,* the tangent to the space circle must be perpendicular to the corresponding N_i It is the need for some means to guarantee this tangent property that leads to the following observations and constructions.

We select a distinguished point *Pi;* without loss of generality assume that it is the point P1. We then construct all the space circles through the translated point *7*1* whose tangents at *T1* are perpendicular to *N1,* and that also pass through a second translated point, call it *T2-* There is a one degree of freedom family of such circles, and they all lie on the surface of a sphere. That such a construction is possible can be seen both geometrically and algebraically; it is also a special case of the theorem of Meusnier. (Meusnier's theorem states that if a set of planes are drawn through a tangent to a surface in a non-zero curvature direction, then the osculating circles of the intersections with the surface lie upon a sphere [Struik, 1961].)

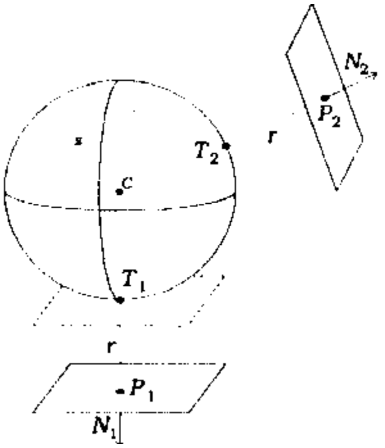

Fig. 2: Minor r extraction geometry.

Geometrically, we consider *T* to be the south pole of a sphere whose north-south axis is collinear with *N1.* The size of the sphere is determined by T2. Call this sphere the supporting sphere. Any circle (great or little) that passes through T1 and T2 now also has its local tangent at T1 perpendicular to N1. Algebraically, we can look for the size of the support sphere, *s;* since we know that the center is constrained to lie along the direction of the *N1,* the center is given by *C=T1+sN1.* Both *T* and *T2* lie on a common sphere if their distances from the center are equal. This is captured by equating the norms of their relative pOsSition vectors, giving in vector form the equation of the plane of their perpendicular bisectors: $|C-T_1|=|C-T_2|$.. Expanding, we find that $s_2(r)$, that is, the size of the support sphere needed to accommodate *T2* ,is a function of *r.*

$$s_2(r) = \frac{2r\,N_2{\cdot}P_{1,2} - P_{1,2}{\cdot}P_{1,2}}{2r\,(1 - N_1{\cdot}N_2) + N_1{\cdot}P_{1,2}}$$

Note, however, that the local tangent at T2 is not necessarily perpendicular to *N2-* Worse, T3 may not even lie on the support sphere at all.

We remedy this second defect first, by constructing with *T* and T3 the same sort of support sphere as was constructed with T1 and T2- What results is a symmetric relationship,

$$s_3(r) = \frac{2r\,N_3 \cdot P_{1,3} - P_{1,3} \cdot P_{1,3}}{2r\,(1 - N_1 \cdot N_3) + N_1 \cdot P_{1,3}}$$

which expresses the size of the support sphere which would not only accommodate T3, but which would also allow us to draw a one-dimensional family of circles through *T1* and *T3* whose local tangents at *T1* are perpendicular *N1.*

Now note that if $s2(r) = s3(r)$, for whatever *r,* the support spheres must be coincident, since they are both defined relative to *P1,* and all three Ti lie upon it. Basically, equating $s2(r) = s3(r)$ will find those *r* that allow a solution to the now overdetermined problem of finding a sphere that passes through three points and attains a specified normal at a given point (here, the south pole). Equating $s2(r) = s3(r)$ gives a quadratic in r; $Ar^2 + Br + C = 0$, where

$$A = 2\,N_2 \cdot P_{1,2}\,(1 - N_1 \cdot N_3) - 2\,N_3 \cdot P_{1,3}\,(1 - N_1 \cdot N_2)$$

$$\begin{aligned} B = {} & 2\,N_2 \cdot P_{1,2}\,N_1 \cdot P_{1,3} \; - \; 2\,N_3 \cdot P_{1,3}\,N_1 \cdot P_{1,2} \\ & - P_{1,2} \cdot P_{1,2}\,(1 - N_1 \cdot N_3) \; + \; P_{1,3} \cdot P_{1,3}\,(1 - N_1 \cdot N_2) \end{aligned}$$

$$C = -P_{1,2} \cdot P_{1,2}\,N_1 \cdot P_{1,3} + P_{1,3} \cdot P_{1,3}\,N_1 \cdot P_{1,2}$$

This equation will have at most two solutions for the value of r, and they are easily obtainable by the quadratic formula. Since they are based on *T1* being the south pole, call them 1,1 and ri,2. We still have to address the need for T2 and T3 to satisfy their tangent condition, however. We do this in a completely symmetric fashion, by finding two more support sphere systems, first by considering T2 a south pole, and then T3. The quadratic equations in *r* that result are derived by inspection by permuting the indices of the *Pi* and *Ni.*

From T2 as south pole we get two more candidates for the value of r, r2,1 and r2,2, and from *T3* we get r3,1 and r3,2. This would appear to call for the calculation of 24 inner products, but it is easy to show that any two of these three quadratics have four inner products in common; thus, there is a total of only 12 innerproducts.

We now have three pairs of candidate *r* values. Let us define a consensus *r* to be any *ri,j* that satisfies

(2a) $$(r_{1,1} \text{or } r_{1,2}) = (r_{2,1} \text{or } r_{2,2}) = (r_{3,1} \text{or } r_{3,2})$$

There may be zero, one, or two consensus *r.* They can be accumulated and filtered in the usual Hough way using a one-dimensional parameter space. We note several properties of this consensus algorithm.

The values of *r* are derived by independently solving three quadratics in 15 image observables (three for each point *Pi,* and two for each unit normal *Ni),* The virtue of this method is that despite the non-linearities that would result when variables are eliminated, the method does not require any iterative root finding. Nor does it suffer from the attendant problems of choosing starting values, guaranteeing convergence, or tracking multiple roots.

Enlisting the aid of IBM's proprietary symbolic math system Scratchpad III [Jenks *et al.,,* 1986], we were able to show that the transform always produced one consensus value for r, corresponding to the true minor radius of the torus, and that if a second consensus existed, it was always easily distinguishable by its sign and/or magnitude from the true consensus r, which is always the smallest positive radius of curvature.

Further, the analysis suggested several intriguing examples of what should probably be called anti-tori. The most straightforward example is what happens when two of the data points are on a meridian, and the third is on the innermost parallel. The transform properly returns two consensus: one is r, and the other is $-1/(a-r)$, the value of k_2 at the "innermost" parallel. What the transform "sees" is an anti-torus whose thickness is equal to the torus' hole, and whose hole is equal to the torus' thickness. That is, it interprets the data to be lying on the negative image of an anti-torus, whose axis is perpendicular to the given torus, with the torus and anti-torus interlocking: much like confusing the impression in plaster of a face with the face itself.

3. Computing Orientation (and Major Radius)

Following the principle of least variability, we next recover orientation by examining the spine points produced by our candidate *r.* Orientation is a naturally bounded quantity, and is easily represented as a point on the surface of the upper Gaussian hemisphere (mathematically, on P^2.

The approach is straight forward: given a candidate r, we form the spine points Si = *Pi - rNi* from the pixels supporting r, and determine the orientation of the plane on which they lie. The direction of the plane's normal is easy to compute: simply take the normalized unit vector of the following cross product (or any of its variants obtained by permuting the *Si):*

$$(S_1 - S_2) \times (S_3 - S_2)$$

However, this orientation is ill-conditioned when the spine points are nearly co-linear, which occurs when *a* is large. Using derivatives, it is not hard to show that small changes of *S2* in a direction perpendicular to this plane have an effect on the plane's orientation that is roughly proportional to the major radius, *a.* This suggests that the computation for orientation should be accompanied by the computation of *a,* so that orientation can be "weighted" inversely by *a* somehow.

Rather than use a as a pure parameter space voting weight, we note the following. The value of *a* eventually has to be recovered anyway, and if it disappears into a weighted sum it is not recoverable. Instead, we scale the unit orientation vector by dividing it by a. The parameter space now becomes the interior of the upper Gaussian hemisphere, which is three dimensional, and the values of *a* are recoverable simply by taking each vector's length. More importantly, orientations accompanied by high *a* cluster near the origin, where they are easy to detect and remove. Conversely, other orientations receive parameter space representation proportional to their certainty, since they are proportionately distant from the hemisphere center.

This parameter space has two other advantages. First, it uses the interior of what would otherwise be a very inefficient use of three-space, thus obviating the need to cleverly tesseilate the surface of the Gaussian sphere. Secondly, points that cluster at its origin can immediately be considered evidence of cylindrical objects in the image (which have no orientation and infinite *a).*

Since the orientation and *a* spaces are now combined, the computation of the resulting parameter point (the scaled orientation vector) can also be combined; it is not necessary to actually plot the spine circle or calculate its center. By means of a three-space geometric construction related to the planar "Law of Sines" for circles, it can be shown that the scaled vector is directly given by the following (or by any of its variants obtained by permuting the *Si):*

$$\frac{2W_1 \times W_3}{|W_1||W_2||W_3|}$$

where

$$W_1=[S_3-S_2],\ \ W_2=[S_1-S_3],\ \text{and}\ W_3=[S_1-S_2].$$

4. Computing Center

What remains are the three most difficult parameters, the center parameters, which are potentially unbounded and highly sensitive to the value of a. Local tori with small curvature have distant centers which are difficult to compute accurately. Since triangulation error of the center is also roughly proportional to a, this final space computes and accumulates vectors to the center point, scaled again by *a.* In effect, this measures each center in units of major radius, thus relative error is nearly constant. It is not hard to show that this space is now bounded, since for local tori with large *a,* their center must be about *a* units from the image origin, otherwise they would not fall within the image. Thus, most tori have centers within one unit from the origin. The upper limit of this space is determined by small tori at the image edge; since *a* is bounded by r, and *r* is bounded by physical considerations, this limit is calculable directly.

We note that although many tori might map to the same scaled vector, they have already been classified and separated by the value of *a* in the prior parameter space. Thus, the nesting of these spaces is critical both for adjustment to error, and for disambiguation of results.

The computation is straightforward but a bit messy (however, it is again invariant to the permutation of the *Si):*

$$C=\frac{aS_1+bS_2+cS_3}{g}$$

where

$$g=D_1D_3-DD;\ \ a=D_1D_3-DD_1;\ \ b=DD_2;\ \ c=D_1D_3-DD_3$$

and

$$D_1=W_1\cdot W_1;\ \ D_2=W_2\cdot W_2;\ \ D_3=W_3\cdot W_3;\ \ D=W_1\cdot W_3.$$

5. System Considerations

In practice, the above method relies on the ability to pick "good" triples of points *Pi.* Additionally, the performance and behavior of the method can be enhanced by simple "sanity checks" on computed intermediate results. Lastly, the orientation space can be improved by a judicious choice of "offset" for the scaled orientation vector. We handle these in turn.

Picking good triples is determined by enforcing a minimum and maximum distance between image points/*/; we call this distance the "radius of coherence" (ROC). If the ROC is too small, accuracy suffers due to small triangulation baseline; too large, and most triples do not lie on the same torus. For the minor radius space, it can be set in accordance with the expected range of minor radii, which can be determined from the imaging parameters. Empirically, we have found that a wide range between minimum and maximum ROC works well for minor r. The two other spaces have similar considerations, however the range between minimum and maximum ROC must be rather narrow or the system becomes overwhelmed with noise hypotheses. Fortunately, experience shows that a narrow range of ROC about *r* works very well for a large range of torus shapes.

Sanity checks are inexpensive checks on the data before it is used in a parameter transform. Our sanity checks are of the following form. In computing r, no points in a planar neighborhood are used; they are easy to detect, since they have very small and equal principal curvatures. Nor are points near a depth discontinuity used, as the surface approximations become inaccurate there. Further, degenerate quadratics (with A=0 or with imaginary roots) are ignored . In computing orientations and centers, spine points within a pixel of each other are ignored. These checks take a minute percentage of the computation times, but reduce the noise in parameter space dramatically.

In practice, the scaling of orientation vectors by 1/u results in vectors too tightly clustered around the origin. This is because any torus large enough to be seen in the image as a torus will have a major radius of at least eight pixels, approximately. Thus, all the activity in the space happens in a hemisphere of radius 1/8; this uses less than 1% of the space. If a lower bound on *a* is known (and it usually can be approximated), then the scaling function should be of the form ^{b}*/(a+c), wi*th *c* serving to shift small values of *a* closer to the surface of the hemisphere. Heuristic choices of constants based on expected torus and image sizes can be selected; one good one maps the smallest torus onto the surface of the hemisphere, and the largest torus that can be fully seen (a=64, assuming a range image of size 256) into the midpoint of the hemisphere; the scaling function becomes ^{56}/(a+48).

6, System Description

We briefly survey the complete system of which this transform is a part, highlighting those aspects most germane to our results. A more complete description can be found in [Kjeldsen *et al,* 1989].

Recognition is structured as a hierarchy of layered and concurrent parameter transforms [Ballard, 1981][Sabbah, 1985]. Each transform examines input data or previously established features and accumulates evidence for new feature hypotheses in an associated parameter space. Compatibility relations accumulate evidence for or against a hypothesis on the basis of peer hypotheses. A large number of hypotheses are typically generated. The evidence for and against each is integrated using an iterative refinement process in a dynamically constructed constraint satisfaction network [Feldman & Ballard, 1981].

Each parameter space is instantiated as a subnetwork where nodes correspond to hypotheses. The links in the network are (1) bottom-up connections, representing support from input data, and (2) lateral links between hypotheses. The former links can be thought of as votes generated by the parameter transforms [Hough, 1962]. The latter links, generated by the compatibility relations, can be inhibitory, or excitatory. For example, surface hypotheses generated from the same pixels are connected by an inhibitory link, because they represent conflicting interpretations of the same data.

Evidence integration works as follows. Each node *i* computes an activation level ui representing a confidence level in the corresponding feature. At each iterative step t, the activation level of a node, denoted by ui(r), is computed as

$$u_i(0)=0$$
$$u_i(t+1)=u_i(t)+I_i+\sum_j w_{ij}\,u_j(t)-D_i\,,$$

where li represents bottom-up support for feature i, and the summation embodies the collective inhibition and excitation of conflicting and cooperating hypotheses;. The weight factor *wij* is negative when hypothesis *j* conflicts with hypothesis i and positive when hypothesis *j* is supporting i. *Di* is a decay term that suppresses spurious hypotheses with little support, and helps insure stability.

A unit "survives" iteration when it has sufficient *ui* and insignificant inhibition. It is then passed to the next parameter transform in order to create hypotheses in higher-level spaces. Units also feed-back to their component features in lower-level spaces and to consistent hypotheses in parallel spaces. Thus, surviving features form stable coalitions which represent globally consistent interpretations of the scene.

Additional features are added to the system by defining parameter transforms and compatibility relations which work within a well defined I/O structure. These generally make use of techniques described in [Califano, 1988] and [Califano *et al,* 1988]. Our first transform takes triples of data points and returns values of *r* on the basis of the consensus in Eq 2a. The second transform takes triples of the data points that support a surviving *r* hypothesis, and returns the parameters of the scaled vector of section 3. Finally data points supporting surviving orientation/major radii are used as described in section 4. The current system contains 21 parameter transforms for various features.

The lowest level of the system extract local features such as surface approximations or depth discontinuities from the data for use by the parameter transforms. The local features important to this work use bicubic interpolations to obtain the least mean square error fit to dense depth data, then computes directly from the polynomial coefficients of the approximation both the surface gradient vector, and the directions and amounts of principal surface curvature [Sabbah & Bolle, 1986]. Experiments comparing computed values with known true values indicate that over a wide range of imagery and circumstance the inaccuracy of the approximations is no more that 5% from the ideal.

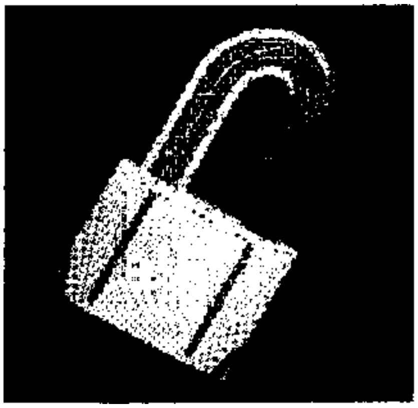

Fig. 5: Points voting for minor radius

7. Results

The torus extraction transforms have been run on about a dozen images. We will present two of the more interesting cases. The first is a depth map generated from a CSG representation of a padlock (figure 3). The image contains a single torus segment, as well as several other surfaces. The second image is an actual range image of a knotted length of cable (figure 4). The cable forms a continuously varying tube of constant cross section, which can be reasonably approximated as piecewise toroidal.

In both cases we used only a small percentage of the possible triples within the ROC; the percentage was large enough to give reasonable coverage but small enough to give acceptable running times. Additionally, hypotheses receiving votes from fewer than 20 triples were not instantiated. Since many noise hypotheses receive just a few votes, this pruning helped cut down on the time and memory needed to support them.

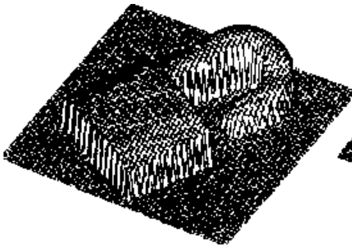

Fig. 3: Depth map of lock.

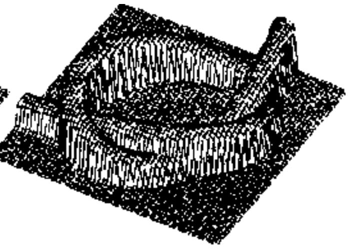

Fig. 4: Depth map of knot.

7.1. Lock

Figure 3 shows a 64x64 plot of the 256x256 depth map of the lock. Surface approximations were taken using a *5x5* window around each point. After the sanity checks, only the points hilighted in the dithered image in figure 5 were passed to the minor radius parameter transform. ROC was 9 to 10, and 50% of the possible triples were used. 10 hypotheses received votes from more than 20 triples, and so took part in iteration. After 11 iteration steps a single hypothesis of 16.5 for *r* survived, which corresponds closely to the apparent radius of the hasp.

Figure 6 shows the points which supported the winning r. To find the orientation/major radius, ROC was set to range from 15 to 16 (approximately *r)* and 50% of the triples formed from the spine points were used. 102 hypotheses received sufficient support to be instantiated.

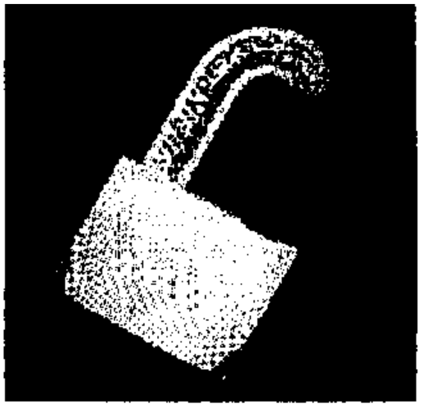

Fig. 6: Pixels supporting winning r.

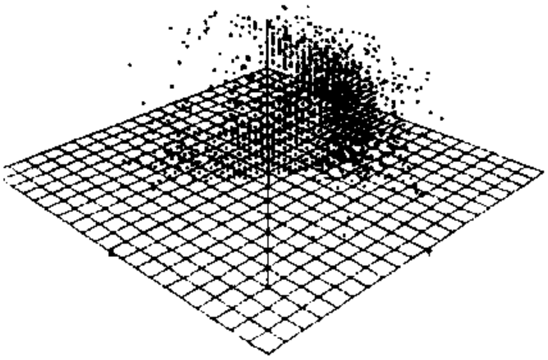

Fig. 7: Orientation hyotheses plotted in 3-space.

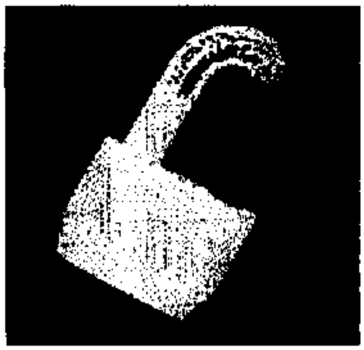

Fig. 8: Pixels voting for orientation 1.

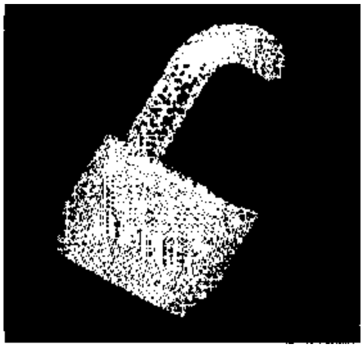

Fig. 9: Pixels voting for orientation 2.

Figure 7 shows the hypotheses receiving votes plotted in 3-space (it does not show the number of votes each received). You can make out the hemispherical outlines of the parameter space, as well as a distinct cluster of hypotheses in the -x +y +z quadrant. Two hypotheses survived after iteration. The points supporting them are shown in figures 8 and 9. The first captured the orientation of the torus forming the top of the hasp as closely as we were able to measure it ourselves. The major radius was captured as accurately as the parameter space resolution (bucket size) would allow (actual a=34, computed a=37). The second surviving orientation hypothesis "found" a very large diameter torus (a=523) on the straight segment of the hasp. As we mentioned in section 3, it is possible to eliminate hypotheses of very large radius, that is near-cylinders by discarding those very close to the origin of the parameter space. Since the length of the parameter vector of the torus on top of the hasp is .65, and the length of the parameter vector for the torus found on the cylinder was .1, the cutoff could easily be adjusted to eliminate such misinterpretations.

Both orientation hypotheses now voted into location space. The same ROC and hit rate were used. The cylindrical segment was not able to find any consistent location, and therefore created no hypotheses strong enough to be considered. The actual torus created 25 hypotheses. Only one survived iteration. The surviving center was (30 67.5 -30), with the actual center at approximately (31 66 -32). Thus, the system found the results to within one bucket of available resolution.

These runs were done on a Symbolics 3650. With the parameters set as described the entire recognition took roughly 70 minutes. Computing the surface approximations took 10 minutes. Generating the votes for minor *r* space took 30 minutes. Generating votes for orientation space took 25 minutes. Generating votes for location space took 3 minutes. In all, 30 iterations were needed to prune down the hypotheses. Total time for all iteration was roughly 3 minutes, including a rather elaborate trace of system status.

7.2. Knot

The second image is a range image of a knot of coax cable (figure 4) taken using a laser triangulation range finder [Technical Arts, 1986]. ROC was the range 4 to 5, hit rate was 50%. Figure 10 shows the points supporting the winning (and correct, as far as we can measure) minor radius hypothesis.

These points were passed thru the orientation/major radius parameter transform. Despite the lack of distinct peaks in the histogram of votes (figure 11), the iterative refinement was able to find three distinct clusters, representing competing hypotheses of orientation from different areas of the knot. The strongest hypothesis in each cluster survived iteration. The points supporting them are shown in figures 13, 14 and 15. The light coverage is due to the low hit ratios (10%) we had to use. The low hit rate also appears responsible for some regions of the knot not being covered by any hypotheses.

Figure 12 shows a histogram of the votes the three orientation hypotheses generated in location space. One peak was generated by the segment in fig. 15, and the other peak is an overlap of the votes from both fig. 13 and fig. 14. The two locations which survived iteration correspond well to the center points we expected from those torus segments.

Running times were somewhat better than those for the lock test case, except for the time to generate votes for orientation hypotheses. Here the continuously varying curve created a very large number of noise hypotheses, and very wide peaks in the histogram of votes. With the 10% hit rate used

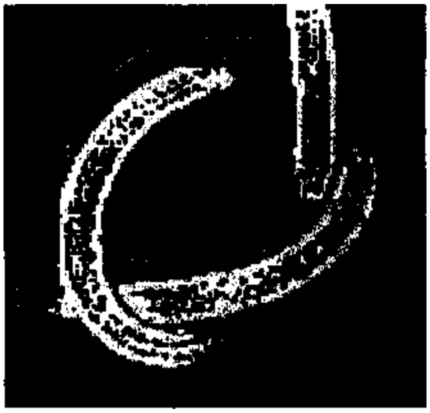

Fig. 10: Pixels supporting winning minor radius.

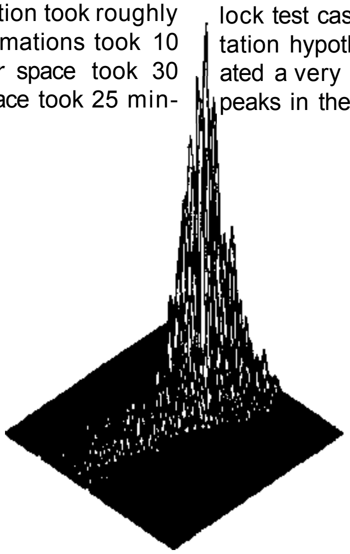

Fig. 11: Knot orientation votes / 2D projection.

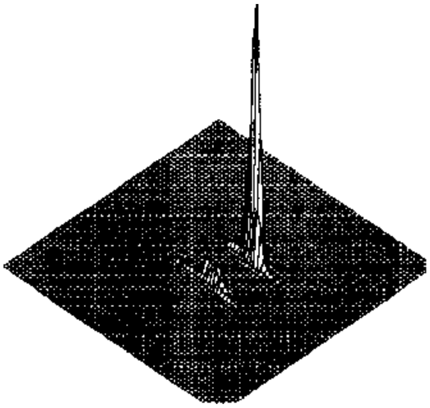

Fig. 12: Pixels supporting winning location.

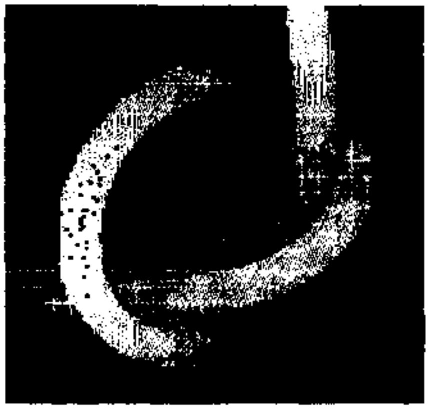

Fig. 13: Pixels supporting orientation hypothesis 1.

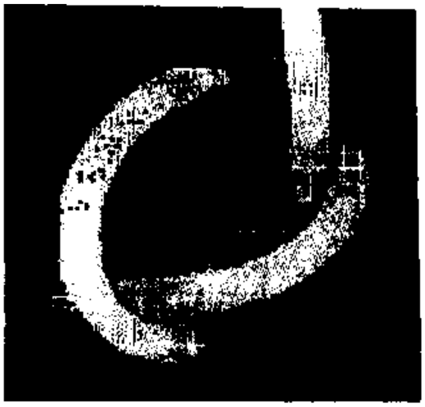

Fig. 14: Pixels supporting orientation hypothesis 2.

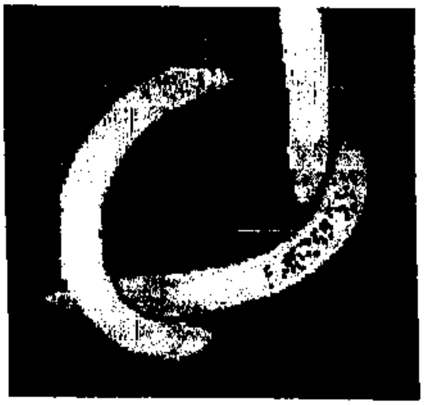

Fig. 15: Pixels supporting orientation hypothesis 3.

here, roughly 4000 hypotheses were created. Linking them took well over an hour. With somewhat higher hit rates, 21000 hypotheses were created. Many hours were needed to create and link them.

7.3. Discussion

These experiments demonstrate the ability to break a continuously varying curve into piecewise toroidal segments, to deal with non-toroidal surfaces and cylinderical segments. We hope that speeding the implementation and a future port to faster hardware will improve the coverage, but these results are very promising.

System parameters must currently be adjusted for each image. An initial setting generally gives either impossibly long running times, due to the number of hypotheses created, or poor coverage of points supporting the winning hypotheses. The latter makes recognition in subsequent parameter spaces difficult. ROC and hit rate can generally be adjusted for good coverage in reasonable time, but it takes several attempts. A faster implementation would allow us to use all triples and a wider ROC, especially in minor radius space. We believe this would almost eliminate the need to tune system parameters.

Acknowledgments

This work was performed while the first author was partially supported on sabbatical leave by IBM Hawthorne; additional support was provided by the Defense Advanced Research Projects Agency, contract N00039-84-C-0165. We thank Sanjaya Addanki, Ruud Bolle, Andrea Califano, and Rakesh Mohan for their generosity and insight. We especially are grateful to another visitor, Russell W. Taylor, for many useful discussions over donuts, and anti-donuts.

References

[Ballard, 1981] D.H. Ballard, "Parameter nets: A theory of low level vision," in *Proc. 7th Int. Joint Conf. On Artificial Intell.,* Aug. 1981, pp. 1068-1078.

[Califano, 1988] A. Califano, "Feature recognition using correlated information contained in multiple neighborhoods," in *Proc. 7th Nat Conf. on Artificial Intell.,* July 1988, pp. 831-836.

[Califano *et ai,* 1988] A. Califano, R.M. Bolle, and R.W. Taylor, "Generalized neighborhoods: A new approach to complex feature extraction," *IEEE Conf on Comp. Vision and Pattern Recognition,* Nov. 1988.

[DoCarmo, 1976] M.P. DoCarmo, *Differential geometry of curves and surfaces.* New Jersey: Prentice-Hall, 1976.

[Feldman & Ballard, 1981] J.A. Feldman and *DM.* Ballard, "Connectionist models and their properties," *Cognitive Science,* Vol. 6, 1981, pp. 205-254.

[Hough, 1962] P.V.C Hough, Methods and Means for Recognizing Complex Patterns, U.S. Patent 3069654, 1962.

[Jenks *et al.,* 1986] R.D. Jenks, R.S. Sutor, and S.M. Watt, "Scratchpad II: An abstract datatype system for mathematical computation," IBM Tech. Rep. RC12327, Nov. 1986.

[Kender & Kjeldsen, 1991] J. R. Kender and R. Kjeldsen, "On Seeing Spaghetti: A Radius-finding Transform for Flexible Extruded Objects," IBM Tech Rep. RC16576, Feb. 1991.

[Kjeldsen *etai,* 1989] R. Kjeldsen, R.M. Bolle, A. Califano, and R.W. Taylor, "A Homogeneous Framework for Visual Recognition," *in Proc. 11th Intl. Joint Conf on Artificial Inteli,* August 1989, pp. 1578-1584.

[Sabbah, 1985] D. Sabbah, "Computing with connections in visual recognition of origami objects," *Cognitive Science,* Vol. 9, No. 1, Jan-March 1985, pp. 25-50.

[Sabbah & Bolle, 1986] D. Sabbah and R.M. Bolle, "Extraction of surface parameters from depth maps viewing planes and quadrics of revolution," in *Proc. SPIE Conf Intell Robots and Comp. Vision,* Oct. 1986, pp. 222-232.

[Struik, 1961] D.J. Struik, *Lectures on Classical Differential Geometry,* Addison-Wesley, 1961.

[Technical Arts, 1986] Technical Arts Corporation, *100X 3D Scanner: User's manual & application programming guide.* Redmond, WA: 1986.