Rational Discrete Generalized Cylinders and their Application to Shape Recovery in Medical Images

James P. Williams[1], John K. Johnstone[2] and Lawrence B. Wolff[1]

[1] Computer Vision Laboratory
Department of Computer Science
The Johns Hopkins University
Baltimore, MD 21218
{jimbo, wolff}@cs.jhu.edu

[2] Graphics and A.I. Laboratory
Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, AL 35294
johnstone@cis.uab.edu

Abstract

Generalized cylinders (GCs) are a popular representational tool in computer vision. In medical imaging, the curved axis GC is particularly applicable to a number of elongated physical structures such as vasculature, bone and bronchi. In many of these instances, it is necessary to recover curved-axis GCs with arbitrary cross-sections. It is also vital that these structures, once recovered, can be analyzed and visualized with off-the-shelf algorithms and software packages. Such tools are usually designed to operate on the domain of polynomial or rational surfaces.

Unfortunately most extant, suitably versatile GC representations do not admit rational parameterizations. We develop an entirely rational B-spline representation for generalized cylinders with curved axes and arbitrary cross-section functions. We demonstrate how our representation can be used as a deformable model by extracting a rational GC from pre-segmented spinal data using a discrete dynamic surface fit.

1. Introduction

The generalized cylinder (GC) has received much attention in the vision literature. It has proven itself as a versatile and reliable representation in a wide variety of applications. Starting with the early models of Binford [2] the GC has appeared in a variety of progressively more expressive incarnations [11, 14, 15]. One drawback of the GC model is that GCs are often difficult to express rational surfaces, or, more specifically, as rational B-spline surfaces. Rational B-spline surfaces are a representational standard of solid modeling and rendering tools. Algorithms for intersections, blends, computation of curvature and countless other manipulations are readily available for B-spline surfaces. Many of these algorithms are already embedded in commercial CAD packages or available in free software libraries.

Some researchers have addressed this problem with algorithms that convert specific types of GCs into B-spline surfaces [3, 4]. There are several drawbacks to using conversion. First, a conversion algorithm relies on a specific GC representation and must be altered for each unique GC incarnation. Second, conversion can only approximate a surface if the underlying GC is inherently not rational. Finally, foremost among our concerns is that conversion is not applicable in an active surface framework [9, 10, 18] because forces need to act on the GC directly, not on some approximate B-spline representation thereof.

When fitting an active surface to empirical data, we must effect many small incremental changes both globally and locally to the surface. Each subsequent iteration of an active surface algorithm relies on information from the previous iteration. This information is often obtained by measuring characteristics and positions of the surface relative to some background field of empirical data. These measurements can involve measuring curvatures or volumes, estimating integrals, calculating intersections and numerous other tasks to which the B-spline representation is well suited. It is natural therefore, to use the B-spline surface itself as an active model.

We consider GCs under the framework of *swept surfaces*. A swept surface is a surface defined by sweeping a planar cross section along a space curve. The swept surface is a modular representation that can be used to describe GCs as a function of three independent components: the axis, the cross-section curve, and the orientation function. The axis curve determines the translation of the cross-section as it is swept. Cross-section functions are closed planar curves defined in a canonical frame. The orientation function describes the rotation of this canonical frame into some time-varying orthogonal frame in space.

When the main axis of the GC is linear, the orientation function is constant. Such GCs are easy to represent as B-spline surfaces. Unfortunately, once a non-linear main axis is introduced, the orientation function is no longer a constant. This has been an impediment to developing rational, curved-axis GCs. Techniques suitable for creating rational B-spline orientation functions have recently been presented in [8]. We adapt these techniques to GCs to produce a representation compatible with the deformable-model

1063-6919/97 $10.00

paradigm.

We are interested in a class of GC which has an arbitrary B-spline space curve as its axis, and varying planar B-spline curves as cross sections. Shani and Ballard [15] developed a similar non-rational type of GC. Their *discrete generalized cylinder* (DGC) had the following properties[1]:

1. *The main axis is a free-form continuous 3-D curve.*
2. *Each cross-section is a closed, free-form, continuous planar curve.*
3. *Cross-sections are perpendicular to the main axis.*
4. *The main axis passes through centroids of the cross-sections (or some approximation thereof.)*
5. *Cross-sectional curves are explicitly defined only at discrete points along the main axis.*

The DGC used a B-spline main axis and cross-section function, but in order to meet condition 3, it was necessary to rotate the planar cross-section using non-rational orientation functions.

The orientation issue has another problematic element. As we mentioned earlier, cross-sections are usually designed in some canonical reference frame and then rotated into a plane perpendicular to the main axis to meet condition 3. This transformation has a degree of freedom that corresponds to rotation of this plane around the tangent to the main axis. An arbitrary choice for this free variable can twist and constrict the resulting GC. The obvious choice of the Frenet frame from differential geometry is not suitable because of this twisting behavior and additional degeneracies.

More recently, O'Donnell et al. [12] address the problem of degeneracies of the orientation function with their *extruded generalized cylinder* (EGC) model. This model avoids singularities of the orientation function but is not a spline-based or rational surface. The EGC formulation also does not provide a minimization for twist, and does not allow arbitrary planar cross-sections.

We derive a completely rational B-spline representation for a curved axis, arbitrary cross-section GC. Our representation, the *rational discrete generalized cylinder* (RDGC) delivers a non-degenerate, B-spline orientation function. In addition, we provide a metric for twist and a procedure for twist minimization. We demonstrate an application of RDGCs in a deformable model framework by fitting a RDGC to pre-segmented spinal CT data.

2. RDGCs as swept surfaces

Consider a closed planar curve $C(s,t)$ deforming and rotating and translating through space as a function of time. Assume this curve is defined in the x-y plane and that the origin is the reference point of the local coordinate system for $C(s,t)$. Let us also assume, w.l.o.g., that the origin is inside the closed curve The translation of the local coordinate system relative to the world coordinate system as a function of time is determined by the main axis curve for the swept surface, $D(t)$. All planar cross section curves are defined in the same local coordinate system that we will denote as:

$$\mathcal{E} = \{\bar{e}_1, \bar{e}_2, \bar{e}_3\} = \{(1,0,0),(0,1,0),(0,0,1)\}$$

In order to enforce condition 3: "*Cross sections are perpendicular to the main axis*", it is necessary to rotate the reference frame $\mathcal{E}$ into an orthonormal frame of the form:

$$\mathcal{F}(t) = \left\{ \bar{v}_1, \bar{v}_2, \frac{D'(t)}{\|D'(t)\|} \right\} \quad (1)$$

where $D(t)$ is the main axis curve. $\mathcal{F}(t)$ is a parameterized frame in which the z-axis points in the direction of the unit tangent vector and the vectors $\bar{v}_1$ and $\bar{v}_2$ span the plane normal to the tangent vector. Given that we have derived such a frame, we denote the rotation matrix that rotates $\mathcal{E}$ into $\mathcal{F}(t)$ as $R(t)$. The expression for the RDGC $G(t,s)$ can now be written as:

$$G(t,s) = R(t)C(s,t) + D(t) \quad (2)$$

The diagram in figure 1 illustrates the role of each of the components in equation (2).

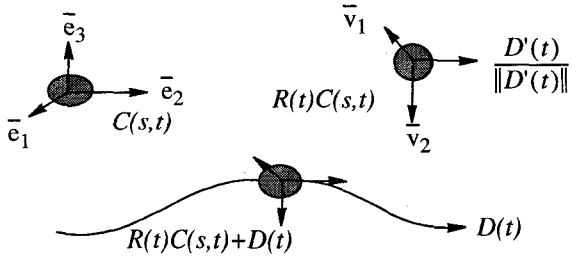

Figure 1. Components of the RDGC and their functions.

3. Rational Orientation Control

There is one degree of freedom in choosing the frame described in equation (1) for any given cross-section. We deal with this design issue in section 5. Let us assume for the moment that we have selected a series of orthogonal frames $(\mathcal{F}_1, .., \mathcal{F}_n)$ and a set of parameter values $(t_1, .., t_n)$ such that at time t_k, we want $R(t_k)$ to be the rotation that takes the reference frame $\mathcal{E}$, of the cross section curve into $\mathcal{F}_k$:

$$R(t_k)\mathcal{E} = \mathcal{F}_k$$

We require that $R(t)$ be a rational function to preserve the overall rationality of our representation. We would also like $R(t)$ to smoothly interpolate between the frames at intermediate time values. This problem becomes easier when cast in terms of quaternion algebra.

3.1. Quaternions and quaternion curves

There are many forms in which rotations can be expressed: Euler angles, rotation matrices etc. The algebraic structure of rotation is best captured in the algebra of Hamilton's quaternions. Quaternions are points on the four-dimensional sphere S^3. Quaternions are the represen-

1. Taken from Shani & Ballard [15] p. 136

tation of choice for smooth interpolation of orientations. One major reason is that the spherical (geodesic) distance between two quaternions on S^3 is the same as the angular distance between rotation matrices.

We express a unit quaternion in the form $q = (s, \bar{v})$ where s is referred to as the scalar component and $\bar{v} \in R^3$ as the vector component. We will not cover quaternion algebra in depth, for further reference see [7] and [16]. It is possible to represent each of the frames $(\mathcal{F}_1, .., \mathcal{F}_n)$ as unit quaternions $(q_1, .., q_n)$. A frame $\mathcal{F}_i$ is represented by the quaternion q_i that rotates the reference frame $\mathcal{E}$ into $\mathcal{F}_i$ at time t_i, i.e. $q_i^{-1}\mathcal{E}q_i = \mathcal{F}_i$. A formula for this conversion is given in Shoemake [16].

A curve embedded in S^3 represents a continuous sequence of rotations. If we construct a continuous rational curve $Q(t)$, embedded in S^3 such that:

$$\forall t_i \in \{t_1, .., t_k\} \qquad Q(t_i) = q_k$$

$Q(t)$ can be used as the orientation curve for an RDGC.

3.2. Rational B-splines on S^3

A solution to the rational interpolation problem on S^3 was introduced in [8]. In this formulation, an invertible rational map onto S^3, $\boldsymbol{M}$ and its inverse $\boldsymbol{M^{-1}}$ are used to reduce the problem of interpolation on S^3 to a general interpolation problem in projective 4-space (P^4). Using $\boldsymbol{M^{-1}}$, it is possible to take quaternions and map them to general points in P^4. These points are then interpolated with a rational B-spline curve. This curve is passed through $\boldsymbol{M}$ to yield a curve on S^3. A three-dimensional sketch of this process is given in figure 2.

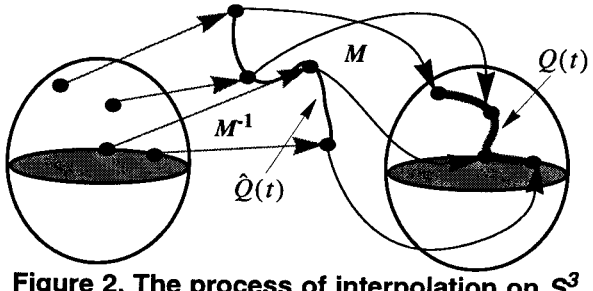

Figure 2. The process of interpolation on S^3 via $\boldsymbol{M^{-1}}$ and $\boldsymbol{M}$.

The map $\boldsymbol{M}$: $P^4 \rightarrow S^3 \subset P^4$ takes a point in projective 4-space to a point on S^3:

$$\boldsymbol{M}(a, b, c, d, 1) = \begin{pmatrix} a^2 + b^2 + c^2 - d^2 \\ 2ad \\ 2bd \\ 2cd \\ a^2 + b^2 + c^2 + d^2 \end{pmatrix} \quad (3)$$

M is a rational map, so if it is applied to a rational curve in P^4, the result is a rational curve on S^3. The restriction that the homogeneous coordinate must be 1 is not problematic as any point in projective 4-space can be re-written as an equivalent point with homogeneous coordinate 1. The inverse of $\boldsymbol{M}$, $\boldsymbol{M^{-1}}$: $(S^3 \subset P^4) \rightarrow P^4$, takes points on S^3 to general points in 4-space:

if $(e \neq a)$

$$M^{-1}(a, b, c, d, e) = (b, c, d, e - a, \sqrt{2(e-a)})$$

otherwise

$$M^{-1}(a, b, c, d, e) = \textit{the hyperplane:}(d = 0)$$

This condition in the inverse indicates that the identity quaternion *(1,0,0,0,1)* has a multitude of inverses. This occurs when the tangent of $D(t)$ points in the direction of $\bar{e}_3$. A slight numerical perturbation of the frame mapping to *(1,0,0,0,1)* will avoid this singularity.

Given these two maps, the algorithm to generate a B-spline curve on S^3 that interpolates $(q_1, .., q_n)$ proceeds as follows:

1. *Apply* $\boldsymbol{M^{-1}}$ *to each* q_k *yielding* $(q_1', .., q_n')$
2. *Homogenize* $(q_1', .., q_n')$ *by dividing through by the 5th component.*
3. *Interpolate* $(q_1', .., q_n')$ *with a general 4-space B-spline* $\hat{Q}(t)$ *with respect to time parameters* $(t_1, .., t_n)$. *(Interpolation methods for B-splines are covered in [1].)*
4. *Generate* $Q(t) = \boldsymbol{M}(\hat{Q}(t))$.

The resulting curve, $Q(t)$, is guaranteed to always lie on S^3 and will interpolate q_k at time t_k. $Q(t)$ is a rational (not polynomial) curve with respect to R^3 because it is defined in projective space. In order to bring the swept surface into the form specified in equation (2), $Q(t)$ can be converted into a rotation matrix.

4. Interpolation of cross-sections

In the RDGC the cross-section function, $C(s,t)$, is not assumed to be a constant function with respect to t. It is a continuous, closed, free-form planar curve. Cross-sections are explicitly defined at a finite number of discrete t-parameter values. We use an approach similar to that used in [15] to generate cross-sectional functions at intermediate values of t.

Currently, we restrict all cross sections to have an equal number of control points. We also require the knot sequence of all cross-sections to be identical. It is possible to define a one-to-one correspondence between the control points of adjacent cross-section functions. For intermediate values of t, we generate control hulls that are linear interpolates of the hulls of the explicitly defined cross-sections preceding and following t.

Given a RDGC $G(t,s)$ with k cross-sections $C(s,t)$, defined at the t values $(t_1, .., t_n)$, the expression for the cross-section function at time t_c: $t_i \leq t_c \leq t_{i+1}$ is a linear interpolate of the cross-sections at times t_i and t_{i+1}:

$$C(s, t_c) = \frac{(t_c - t_i)C(s, t_{i+1}) + (t_{i+1} - t_c)C(s, t_i)}{(t_{i+1} - t_i)} \quad (4)$$

Linear interpolation assures that the RDGC will exactly interpolate the cross-section curve defined at any of the

discrete time steps t_i. However, it also causes the surface of the generalized cylinder to have only C^0 continuity where t is exactly equal to one of the discrete time step values $(t_1, .., t_n)$. C^k continuity can be achieved with higher degrees of interpolation.

5. Choosing frames

Recall from section 1 that in equation (1) there is one degree of freedom in the choice of $\bar{v}_1$ and $\bar{v}_2$. These vectors are free to rotate about the unit tangent vector of $D(t)$ provided that they remain orthogonal. For those familiar with the differential geometry of curves, obvious choices for these vectors are the normal $\bar{n}$, and binormal $\bar{b}$, of the Frenet frame [6]. The Frenet frame is an orthogonal frame based on the first and second derivatives of $D(t)$:

$$\bar{t} = \frac{D'}{|D'|} \qquad \bar{n} = \frac{(D' \times D'') \times D'}{|D' \times D''||D'|} \qquad \bar{b} = \frac{D' \times D''}{|D' \times D''|}$$

O'Donnell et al. [12] pointed out some severe limitations associated with the Frenet frame. The most serious limitation appears when the second derivative of $D(t)$ is identically 0 (as is the case for inflections or along linear portions of curves). In this case the two off-tangent axes of the frame, $\bar{n}$ and $\bar{b}$, are undefined. O'Donnell et al. suggest replacing the Frenet frame with a less problematic formulation. The alternative they propose is a physically motivated analog of extrusion.

The EGC does not use the second derivative of the curve to generate its orthogonal trihedron. Instead, a guaranteed orthonormal frame is generated at each point on the curve by projecting a vector in a "privileged" up direction onto the plane normal to the tangent vector. A third, mutually orthogonal vector is then generated via the cross product. We found the EGC frame was not well suited to RDGCs because it exhibited too much twist around the unit tangent vector. When deforming a RDGC it is desirable to minimize the difference in orientation between consecutive points. This minimizes an undesirable shape property, constriction.

5.1. Twist and constriction

The RDGC is initialized with the initial cross-section functions spaced at regular spatial and parameter intervals. The t parameter varies along the main axis, and the s parameter circles the cylinder. Isoparametric curves of the form $s=k$ are lines parallel to the main axis. Isoparametric curves where t is held constant are circles that lie in planes normal to the main axis.

As the RDGC deforms, these isoparametric relationships are maintained. The $t=k$ curves remain planar and the $s=k$ curves are offset curves of the main axis. If, during deformation, a large degree of twist is introduced between two adjacent cross-sections, a constriction of the surface will occur between those two curves (figure 3).

After the main axis has been determined, it is desirable to minimize the amount of rotation around the main axis

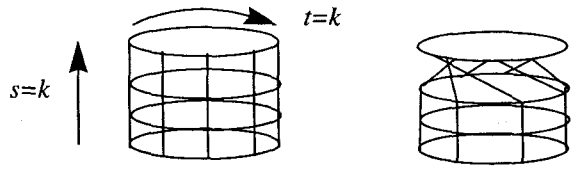

Figure 3. Initial configuration of the RDGC and effects twisting on inter-cross-section correspondences.

between successive frames in the frame sequence. Minimizing the amount of rotation between each such pair limits its constriction and provides rigid invariance of the representation.

5.2. Measuring and minimizing twist

Before we can minimize twist, we must be able to measure it. The metric we use is total rotation. Earlier we demonstrated the equivalence of orthonormal frames to rotations. Any general 3×3 rotation matrix R can be represented as a single rotation by an angle θ about an axis $\bar{v}$. θ is the amount of total rotation performed by R. Quaternions encode rotations in this way. A rotation around the axis $\bar{v}$ by angle θ corresponds to the quaternion:

$$\left(\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\bar{v}\right)$$

The total rotation between two matrices R_1 and R_2 is the total rotation of $R_2 R_1^T$, the matrix that transforms R_1 into R_2. Total rotation has an intuitive geometric definition in the context of the quaternion sphere.

Geodesic distance (length of the minimal great arc) between two points q_1 and q_2 on the quaternion sphere is inversely proportional to the magnitude of the inverse cosine of their inner product $(q_1 \bullet q_2)$. We represent a fixed frame, $\mathcal{F}$ as the unit quaternion that rotates the principal frame $\mathcal{E}$ into $\mathcal{F}$. The family of quaternions that rotates a given unit vector $\bar{v}_1$ into another unit vector $\bar{v}_2$ forms a great circle on S^3.

We fix the first frame in our frame sequence, $\mathcal{F}_1$, to be the Frenet frame of $D(t)$ at parameter value 0. This frame is then converted into a quaternion q_{ref}. The next frame in the sequence, $\mathcal{F}_2$, is calculated based on $\mathcal{F}_1$. First we determine the unit tangent vector $\bar{t}$ of $D(t)$ at $t=t_2$. Let q_a be any one of the quaternions that rotates the z-axis into $\bar{t}$ (this is one of the points on the great circle representing this particular family of rotations.) The parametric equation for the great circle on S^3 that contains all rotations that take the z-axis ($\bar{e}_3$) into $\bar{t}$ is:

$$C = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\bar{t}\right] q_a$$

This equation represents a rotation of the z-axis into t and then a subsequent rotation about $\bar{t}$ by θ degrees. We wish to find θ that maximizes the inner product

$(C \bullet q_{ref})$. Differentiating we find:

$$\frac{d}{d\theta}(C \bullet q_{ref}) = \frac{dC}{d\theta} \bullet q_{ref}$$

After grouping constant coefficients of the variable terms,

$$\frac{dC}{d\theta} \bullet q_{ref} = A\sin\frac{\theta}{2} - B\cos\frac{\theta}{2}$$

solving for θ yields: $\theta = 2\,\mathrm{atan}\frac{B}{A}$

The images in figure 4 show the same portion of a swept helical RDGC with circular cross-sections. Different frame selection methods and their effects on twist are illustrated.

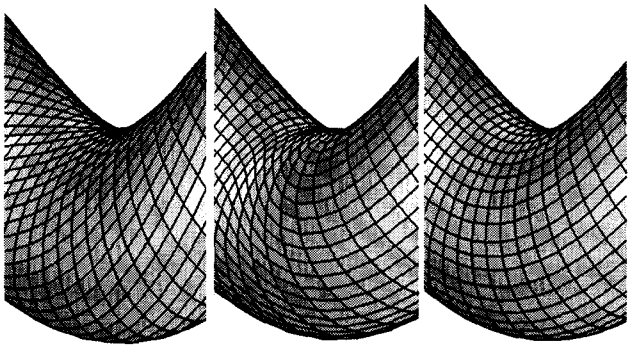

Figure 4. Close-up view of twisting in a helical RDGC defined using Frenet, EGC and minimal twist frames respectively.

6. Incremental deformation

In order to use the RDGC as a deformable model, we must be able to perform numerous small, incremental changes to the surface over a large number of iterations. After every step, the RDGC should conform to the requirements stated in section 1. We examine each one of these restrictions individually.

1. The main axis is a continuous, free-form R^3 curve.

This requirement is satisfied implicitly if the main axis is maintained as a B-spline.

2. Each cross-section is a closed, free-form, continuous planar curve.

Our application, uses closed, C^2 continuous B-spline curves. Since the cross-section curves must always be planar, the forces that act upon them, both internal and external must be applied in the plane or to the plane as a whole. Deformations that warp the cross-section into a general R^3 curve would destroy the RDGC representation.

3. Cross-sections are perpendicular to the main axis.

As the main axis deforms through time, its tangent orientations at the discrete parameter values $(t_1, .., t_n)$ also change. These changes in turn affect the orientation function $O(t)$. A new set of quaternions $(q_1, .., q_n)$ representing the orientations at $(t_1, .., t_n)$ must determined and re-interpolated on the S^3. Therefore we need an efficient method to recalculate $O(t)$ given an incremental change in $(q_1, .., q_n)$.

Let $O_i(t)$ be the new orientation function we wish to calculate and let $O_{i-1}(t)$ be the orientation function of the previous iteration. If the changes made to the orientations during a single time step are small, the curve $\hat{O}_{i-1}(t)$ will be quite close to the new interpolate we wish to derive, $\hat{O}_i(t)$. Instead of constructing a new interpolate from scratch, we use an incremental method for interpolation that performs small displacements of the control points of an initial B-spline "estimate" to quickly converge on a solution [19].

4. The main axis passes through centroids of the cross-sections (or some approximation thereof.)

The main axis is subject to changes in the shape of the cross-section functions. Fortunately, small deformations of an individual planar cross-section have little effect on the position of its center of mass. We use the same type of incremental re-interpolation method for the main axis as we used for the orientation function.

7. Experimental results

We tested the RDGC using a three dimensional modification of the discrete dynamic contour (DDC) model of Lobregt and Viergever [10]. We created a three-dimensional mesh of the RDGC control points by connecting control vertices of equal knot values in adjacent cross-sections. In this way we were able to base the deformation forces on differential properties (normals and curvature) of the curved surface of the RDGC itself.

DDC deformation was implemented as a numerical time integration process. The total force acting on a vertex $\bar{p}_i$ in this model is computed as the weighted sum of internal and external forces plus an additional velocity-based damping force:

$$\bar{f}_i = w_{ext}\bar{f}_{ext_i} + w_{int}\bar{f}_{int_i} + w_{damping}\bar{v}_i(t) \quad (5)$$

The position at time $(t + \Delta t)$ of a vertex $\bar{p}_i$ is a function of its position at time w and its displacement derived from equation (5) yielding the following finite-difference scheme [10]:

$$\bar{p}_i(t + \Delta t) = \bar{p}_i(t) + \bar{v}_i(t)\Delta t$$

$$\bar{v}_i(t + \Delta t) = \bar{v}_i(t) + \bar{a}_i(t)\Delta t$$

$$\bar{a}_i(t + \Delta t) = \bar{f}_i(t + \Delta t)$$

The internal forces we used modeled a reaction-diffusion based involution of the a curve [17]. In a reaction-diffusion model, a surface point translates through time along the direction of its normal vector at a rate which is dependent on a surface curvature term, K, at that point.

Consider an RDGC G evolving with respect to a simulation time parameter w. We consider the derivative of G with respect to the parameter w, this represents the instantaneous velocity of each point on the surface.

$$\frac{\partial G}{\partial w} = (\beta_0 - \beta_1 K)\bar{n}$$

Assuming an outward-pointing convention for surface

normals, a positive value for β_0 will expand the surface like a balloon, while a negative value will cause contraction. The effect of the value β_1 is dependent on the choice of K. Various choices for K are described in [17]. We let β_0 and β_1 be equally weighted at (-1). Initially we ignored whole surface curvature and let K be the normal curvature κ of the cross-section curve. In order to preserve planarity of contours, both internal and external force vectors were projected into the plane of the control point to which they were applied before being used as displacements.

Our goal was to recover the bone surface of a pre-segmented stack of 20 slices of coarse-sampled (1 cm) spinal CT data. The main axis was fixed to an approximation of the medial axis of our data by constructing an interpolate of the individual slice centroids. The orientation function was subsequently computed using the minimal twist method given in section 5.

The RDGC used for the fitting had 20 cross-sections spaced at even parameter values. Each cross-section was initialized to a C^2 continuous B-spline circle with 30 control points and a radius equal to twice the maximum extent of the target data (the distance between the first and last slice of the data.) Four snapshots of the RDGC converging to the data are shown in figure 5.

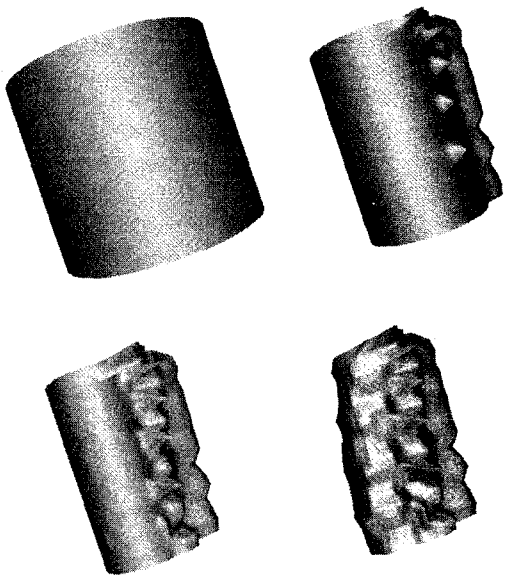

Figure 5. Four steps in the convergence of an RDGC onto coarse spinal column data.

8. Conclusions and future work

A rational representation for curved-axis generalized cylinders was developed. We addressed issues in their design and their applicability in an active surface framework. In addition, we motivated and derived a technique to orient cross-sections in a manner that minimizes twist between the cross sections. The effectiveness of the RDGC as a deformable model was demonstrated on empirical medical data.

Acknowledgments

This research was supported in part by ARPA grant DAAH04-94-G-0278, AFOSR grant F49620-93-1-0484 and the NSF National Young Investigator award IRI-9357757.

References

[1] Bartels, R., Beatty and J. Barsky B. *Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, LosAltos, 1987.

[2] Binford, T. "Visual Perception by Computer." *IEEE Conference on Systems and Control*, Miami, 1971.

[3] Bronsvoort, W. and Waarts, J. "A Method for Converting the Surface of a Generalized Cylinder into a B-spline Surface." *Computers and Graphics*, Vol. 16, No. 2, 1992, pp. 175-178.

[4] Coquillart, S."A Control-Point Based Sweeping Technique." *IEEE CGA*, Nov. 1987, pp. 37-45.

[5] Dickinson, S. and Pentland, A. "From Volumes to Views: An Approach to 3-D Object Recognition." *CVGIP*, Vol. 55, No. 2, 1992, pp. 130-154.

[6] Do Carmo, M. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood, 1976.

[7] Ebbinghaus, H.-D. et al. *Numbers*. Springer Verlag, New York, 1990.

[8] Johnstone, J. and Williams, J. "A Rational Model of the Surface Swept by a Curve." *Proceedings of Eurographics '95, Computer Graphics Forum*, Vol. 14, No. 3, 1995, pp. 77-88.

[9] Kass, M., Witkin, A. and Terzopoulos, D. "Snakes: Active Contour Models." *IJCV*, Vol. 1, No. 4, 1987, pp. 321-331.

[10] Lobregt, S. and Viergever, M. "A Discrete Dynamic Contour Model." *IEEE Transactions on Medical Imaging*, Vol. 14, No. 1, 1995, pp. 12-24.

[11] Nevatia, R. and Binford, T. "Description and Recognition of Complex Curved Objects." *Artificial Intelligence*, Vol. 8, No. 1, 1977, pp. 77-98.

[12] O'Donnell T., Boult, T., Fang, X. S. and Gupta, A. "The Extruded Generalized Cylinder: A Deformable Model for Object Recovery." *CVPR* 1994, pp. 174-181.

[13] Ponce, J. "Straight Homogeneous GCs: Differential Geometry and Uniqueness Results." *IJCV*, Vol. 4, 1990, pp. 79-100.

[14] Shafer, S. and Kanade, T. "The Theory of Straight, Homogeneous Generalized Cylinders." *Technical Report CS-083-105*, Carnegie Mellon University, 1983.

[15] Shani, U. and Ballard, D. "Splines as Embeddings for Generalized Cylinders." *CVGIP*, Vol. 27, No. 2, 1984, pp. 129-156.

[16] Shoemake, K. "Animating Rotation with Quaternion Curves." *SIGGRAPH*, Vol. 19, No. 3, 1985, pp. 245-254.

[17] Tek, H.and Kimia, B. "Volumetric Segmentation of Medical Images by Three-Dimensional Bubbles." *Proceedings of PBMCV* 1995, pp. 1-16.

[18] Terzopoulos, D. and Vasilescu, M. "Sampling and Reconstruction with Adaptive Meshes." *CVPR* 1991, pp. 70-75.

[19] Yamaguci, F. *Curves and Surfaces in Computer Aided Geometric Design*. Springer Verlag, 1988.