Fitting Manifold Surfaces To 3D Point Clouds

Cindy M. Grimm*
Computer Science Dept.
Washington University
St. Louis, Missouri 63130
Email: cmg@cs.wustl.edu

Joseph J. Crisco†
Department of Orthopaedics
Brown University School of Medicine
Providence, Rhode Island, 02912
Email: Joseph_Crisco@Brown.edu

David H. Laidlaw‡
Computer Science Dept.
Brown University
Providence, Rhode Island, 02912
Email: dhl@cs.brown.edu

March 6, 2002

Abstract

We present a technique for fitting a smooth, locally parameterized surface model (called the *manifold* surface model) to unevenly scattered data describing an anatomical structure. This data is acquired from medical imaging modalities such as CT scans or MRI. The manifold surface is useful for problems which require analyzable or parametric surfaces fitted to data acquired from surfaces of arbitrary topology (e.g., entire bones). This surface modeling work is part of a larger project to model and analyze skeletal joints, in particular the complex of small bones within the wrist and hand. To demonstrate the suitability of this model we fit to several different bones in the hand, and to the same bone from multiple people.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling, Surface

Additional Keywords: CT scan, point clouds, least-squares, arbitrary topology

*Supported by NSF grant CCR-992009
†Supported in part by NIH grant AR44005
‡Supported by NSF grant CCR-0093238

Introduction

Digital anatomical structures extracted from medical images are finding a wide range of scientific and clinical applications ranging from finite element modeling to visualization to computer assisted surgery. An important aspect of this is the extraction of surface models of anatomical structures. Our work was stimulated in part by the need to study *in vivo* skeletal joint mechanics. We use these extracted surfaces to quantify joint kinematics, ligament strains, and distances between joint surfaces in normal, pathological, and surgically reconstructed joints [1] [2] [3]. Quantifying these effects requires a surface model which is smooth, locally parameterized, and capable of modeling surfaces of arbitrary topology.

The manifold surface model [4] meets these requirements. First, manifolds are locally parameterized, with the parameterization, and corresponding degrees of freedom, under the control of the user. Therefore, it is a simple matter to provide more degrees of freedom in areas of higher curvature. Second, the fitting process is hierarchically layered, i.e., there is a natural method for doing a coarser to finer fit; the finest fit level is local and is only performed where there is sufficient data to do so. These two properties help with interpolating sparse data since the coarser fit can be used in areas with few sample points and the finer fit applied only where needed. Manifolds are capable of modeling surfaces of arbitrary topology, so we can model entire bones, including those with topological holes and boundaries caused by incomplete data. Finally, the surface is C^k for any desired k, which results in smooth distance-to-surface calculations.

Previous work

There are many techniques for scattered data interpolation; for a recent survey see [5]. We focus here on those which can handle arbitrary topological surfaces of C^1 (or higher) continuity: Spline surfaces [6] [7] [8] [9], algebraic surfaces [10], subdivision surfaces [11] [12] [13], and radial basis functions or thin-plate splines [14].

Of these, all but [8], [14], [13], and [9] require a polyhedron to fit to and produce a single patch per face. It is not clear how well these techniques will work for unevenly scattered points since they provide no mechanism for smoothing or filling holes in the data. They also produce a large number of patches.

The techniques of Eck [8], Hoppe [13] and Krishnamurthy [9] produce approximating surfaces, the first two by simplifying the mesh to produce a coarse network, the last one by having the user draw patch boundaries. These approaches are closest to ours in spirit; the major difference lies in the structure of the output surface. We produce the same local parameterization, in a topological sense, for a single bone across multiple people. Spline patch techniques must also fit both to the interior control points and maintain constraints across boundaries between patches, a notoriously difficult problem with unevenly distributed data.

In [14] the techniques of radial basis functions are extended to handle arbitrary topology. This approach provides smoothing and can handle unevenly scattered data, however the topology of the final surface is not guaranteed to be the same as the input data.

Several approaches specific to modeling joint surfaces exist, such as Boyd's thin-plate splines [15] and Ateshian's B-splines [16], further developed in [17]. These techniques focus on modeling just a single contact region of the bone (topologically a plane). The dynamics of multiple joint surfaces on a single bone make a model of the entire bone (topologically a sphere) more useful than a model of just the contact region for one neighboring bone. Sherrer in [18] produces a collection of patches with enforced C^1 continuity across boundaries. They provide a complete model but only C^1 continuity and also have some difficulties enforcing boundary constraints.

Input data and surface type

The data sets are outer cortical bone surfaces extracted from sequential slices of a CT image volume. The segmentation procedures involved thresholding, image algebra, and user interaction to define each bone contour [1]. This produces dense (≈.01mm) samples along widely spaced (1mm) cross sections.

The manifold surface is described fully in [4]. The topology and rough geometry are specified by a *generator polyhedron*, which specifies the topology and an initial approximate geometry for the higher resolution *manifold polyhedron*, which in turn specifies the connectivity and an initial geometry for a set of overlapping, glued-together spline patches (see Figure 1). The generator polyhedron is constructed by the user and can be any general polyhedron. Each level provides more degrees of freedom than its predecessor.

We use a different embedding equation than the one described in the paper. Our embedding is simpler and also pulls the division by the sum out of the individual patch equations. Each chart is embedded using a single NUBS [19] spline patch E_c and the result blended together using the original blend functions [4].

$$E(p) = \sum_c B_c(p) E_c(M(\alpha_c(p)))$$

where M is either the identity function (vertex and face charts) or the linear transform that takes the domain of the edge chart to the unit square (see Appendix A of [4]). The patch domains for the face charts extend beyond the chart's domain. The control points are placed on the subdivision surfaces as originally described.

The fitting process

The manifold is fit to the data in three steps (see Figure 1). We assume that the user has already constructed a generator polyhedron that has the same topology shape as the data and approximately the same geometry. This takes about an hour and the resulting generator polyhedron can be used for all bones of the same type

(e.g., all hamate bones have the same generator polyhedron). The fitting process brings the approximate geometry into alignment with the data points; it does not change the topology of the manifold surface.

We first find the best fit for the generator polyhedron with default positions for the manifold polyhedron vertices and spline patches. The second step adjusts the vertices of the manifold polyhedron while using the default spline patch locations. Finally, the control points of the spline patches are adjusted if there are sufficient data points that project to that patch's region of influence. Because the surface is approximately correct after fitting the manifold polyhedron, we do not need to fit patches in regions containing few or no samples. Also, we do not have to apply additional constraints to make the boundaries of the patches behave because the patches are overlapped, so boundaries of the patches do not affect the final geometry.

The fitting mechanism

In general, fitting can be expressed as the solution to the following minimization problem, where S is the surface and the d_r are the data points we fit to:

$$\min \left[\Big(\sum_r \min_t (d_r - S(t))^2 \Big) + \alpha_c E_c \right] \tag{1}$$

A point on the surface is described by $S(t)$ for some parameter value t. This equation minimizes the distance between every data point and its closest point on the surface. Additional constraints (the E_c) can be added to enforce "smooth" surfaces, i.e., ones which do not undulate unduly between the given points. The constant α_c expresses the importance of fitting the data versus producing a smooth surface. The curvature term E_c has two purposes. First, it filters noisy data. Second, if the data is very non-uniformly sampled, it serves as a guide for the behavior of the surface in very sparse areas.

Equation 1 requires a non-linear solver. However, if the surface is reasonably well aligned with the data we can form a similar, linear expression by projecting the data points onto the surface and minimizing the

resulting equations. By reasonably well aligned we mean that the closest point on the surface defines a bijection between the surface and the data points with no folds.

The linear expression is as follows:

$$\min \big(\sum_r (d_r - S(t_r))^2 \big) \tag{2}$$

where $S(t_r)$ is the point on the surface closest to d_r. A given point of S can be expressed as $S(p) = \sum_i x_i a_i(t)$ where the x_i are either the polyhedra vertices or the spline control points and the $a_i(t)$ are the blend functions, which, for a given t, evaluate to a constant. After some manipulation, minimizing S reduces to a set of linear equations with the x_i as variables. Specifically, each data point d_r produces an equation of the form $\sum_i x_i a_i^r(t_r) = d_r$, which constrains the surface at t_r to pass through the point d_r for each data point.

The linear optimization problem can be written as a solution to the matrix equation $Ax = d$, where $A = \{a_i^r(t_r)\}$, $x = \{x_i\}$ is the vector of variable and each row r of A corresponds to a linear constraint on x imposed by the data point d_r. We solve for x using a standard least-squares solver for a linear system [20].

Note that the linear approximation, Equation 2, differs from Equation 1 in that we minimize the distance to a specific parameter point t on the surface, not to any point on the surface. We therefore may not find the globally optimal solution.

It remains to show how to calculate the a_i for a specific parameter point t. We begin at the patch level and work up to the generator polyhedron level. The degrees of freedom (the x_is) will be different at each level, and hence so will the matrix A. At the patch level the x_i are the control points of all of the patches. An individual spline patch P_k is of the form $P_k(p) = \sum_i b_i(p) g_i^k$. The entire collection of the control points is therefore $\{g_i^k\}$. Our surface is constructed by "gluing" these individual patches together, i.e.,

$$S(t) = \sum_k \beta_k(t) P_k(t) = \sum_k \beta_k(t) \sum_i b_i(t) g_i^k(t) \tag{3}$$

where the β_k are the blend functions, one for each patch. The blend functions have the property that $\sum_k \beta_k(t) = 1$, with at most three functions non-zero.

Instead of solving for all of the degrees of freedom at once, producing a large but sparse matrix, we take advantage of the overlapping structure and fit patches individually. One way to ensure that $S(t_r) = d_r$ is to ensure that, for every overlapping patch P_k, $P_k(t_r) = d_r$. The resulting error will be at worst the maximum of the individual patch area, since the final surface is a linear combination of the given points. For each patch we find those data points that project onto the domain of that patch and fit to just those points.

For the manifold polyhedron, our degrees of freedom are the vertices of the polyhedron. Each control point g_i^k in its *default* location is expressed as a linear combination of the vertices of the manifold polyhedron, V_j, i.e., $g_i^k = \sum_j B_j^k V_j$. Each point on the surface is therefore of the form:

$$S(p) = \sum_k \beta_k(p) P_k(p) = \sum_k \beta_k(p) \sum_i b_i(p) \sum_j B_j^k V_j \tag{4}$$

Similarly, we can express each vertex of the manifold polyhedron, in its default location, as the sum of the vertices of the generator polyhedron.

Curvature constraints

An advantage of the least-squares formulation is that it does a good job of approximating noisy data. Also, as demonstrated, the spline fitting problem can easily be approximated as a linear problem [21]. The least-squares formulation does, however, behave badly when the weights on a variable are close to zero (visually, this produces "spiking" in the surface). We address this problem by adding additional constraints (the E_c),

which ensure that every variable has sufficient weight. When parameter α_c from Equation 1 is set to one, then the constraint has equal weight to moving a data point an equivalent distance from the surface.

Polyhedral constraints

When fitting the generator polyhedron and manifold polyhedron we add in additional constraints that require vertices to lie at the centroid of their neighbors. If $\{\nu_i\} \subset \{\nu_k\}$ are the n manifold polyhedron vertices forming the star of ν (i.e., all the adjacent vertices) we add the constraint $\beta(\nu - \frac{1}{n}\sum_i \nu_i) = 0$.

Additional patch constraints

When fitting the patches we add in additional constraints at regularly spaced intervals in the parameter space, resulting in a more even support.

We take a uniformly distributed set of points in the domain (a 5 by 5 grid of points) and determine where the embedded points should go based on nearby sample points. Each new constraint finds four data points which surround the embedded point, if any exist, and interpolates between those four data points to produce the desired location for the new constraint.

To find the four surrounding data points we first project all of the nearby data points onto the tangent plane at the constraint point (see Figure 2). Second, we find the four closest points (measuring distance in the plane) such that the four points are "around" the constraint point, i.e., the normalized dot products of the projected points are bigger than 0.5. We then take the weighted average of those point's locations.

Results

For Figure 3 we used one person's scan in the neutral position and the hamate bone from three subjects. We compare our models to meshes created using Nuages [22] software. The sample points we used are the

vertices of the Nuages' meshes; we did not use the Nuages' surface connectivity information. The models ranged in size from 921 to 4754 points. The data points were spaced approximately $0.01mm'$ apart along the contours, with $1mm$ spacing between the contours. On average, the average distance from the data points to the manifold surface is $0.053mm$ with a ± 0.02 95% confidence interval. The average maximum distance was 0.42.

The parameter α_c produces similar results over a wide range of values (from 0.5 to 1.5) except for a few bones with spurious data points on the inside of the bones. For these bones, the larger value of α_c was required.

Discussion

One drawback of the least-squares fitting technique is that uses the closest point to chose the parameter point; if the initial surface and the data points are misaligned this can cause folding or pinching of the surface. This problem is most obvious when the data set has two parallel surfaces close together. One solution is to adjust the generator polyhedron, but this is not very satisfactory.

Conclusion

We have demonstrated a hierarchical technique for fitting a smooth surface to an entire bone. Manifold surfaces have several desirable properties, such as smoothness and arbitrary topology, which make them useful for *in vivo*, multiple joint analysis. The technique requires a minor amount of user interaction for each new bone type, but the remainder of the process is completely automatic. The fitting process is robust in the presence of noise and unevenly sampled data points.

References

[1] Cirsco, J. J., McGovern, R., and Wolfe, S., 1999, "A Non-invasive Technique for Measuring in Vivo Three-dimensional Carpal Bone Kinematics," *J. Orthopaedic Research*, **17:1**, pp. 96–100.

[2] Cirsco, J. J. and Neu, C., 2000, "In Vivo Scaphoid, Lunate and Capitate Kinematics in Flexion-extension," *J. Hand Surg*, **25A:5**, pp. 860–869.

[3] Pike, S., Hulsizer-Galvin, D., Weiss, A., Akelman, E., and Wolfe, S., 2001, "Carpal Flexion/Extension Kinematics are Abnormal in Both Wrists of Patients with Unilateral Scaphoid-Lunate Ligament Tears.," *47th Annual Meeting, Orthopaedic Research Society.*

[4] Grimm, C. and Hughes, J., July 1995, "Modeling Surfaces of Arbitrary Topology using Manifolds," *Computer Graphics*, **29**. Proceedings of SIGGRAPH '95.

[5] Lodha, S. K. and Franke, R., 1999, "Scattered Data Techniques for Surfaces," *Scientific visualization (Proc. Dagsthuhl)*, pp. 181–222.

[6] Peters, J., 1995, "C^1 surface splines," *Siam J. Numer. Anal.*, **32:2**, pp. 645–666.

[7] Loop, C., July 1994, "Smooth Spline Surfaces over Irregular Meshes," *Proceedings of SIGGRAPH 94*, pp. 303–310. ISBN 0-89791-667-0. Held in Orlando, Florida.

[8] Eck, M. and Hoppe, H., August 1996, "Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type," *Proceedings of SIGGRAPH 96*, pp. 325–334. ISBN 0-201-94800-1. Held in New Orleans, Louisiana.

[9] Krishnamurthy, V. and Levoy, M., August 1996, "Fitting Smooth Surfaces to Dense Polygon Meshes," *Proceedings of SIGGRAPH 96*, pp. 313–324. ISBN 0-201-94800-1. Held in New Orleans, Louisiana.

[10] Bajaj, C., Ihm, I., and Warren, J., October 1993, "Higher-order Interpolation and Least-squares Approximation Using Implicit Algebraic Surfaces," *ACM Transactions on Graphics*, **12:4**, pp. 327–347. ISSN 0730-0301.

[11] Halstead, M., Kass, M., and DeRose, T., August 1993, "Efficient, Fair Interpolation Using Catmull-Clark Surfaces," *Proceedings of SIGGRAPH 93*, pp. 35–44. ISBN 0-201-58889-7. Held in Anaheim, California.

[12] Zorin, D., Schröder, P., and Sweldens, W., August 1996, "Interpolating Subdivision for Meshes with Arbitrary Topology," *Proceedings of SIGGRAPH 96*, pp. 189–192. ISBN 0-201-94800-1. Held in New Orleans, Louisiana.

[13] Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., and Stuetzle, W., July 1994, "Piecewise Smooth Surface Reconstruction," *Proceedings of SIGGRAPH 94*, pp. 295–302. ISBN 0-89791-667-0. Held in Orlando, Florida.

[14] Turk, G. and O'Brien, J., August 1999, "Shape Transformation using Variational Implicit Functions," *Computer Graphics (Proceedings of SIGGRAPH 99)*, **33:2**, pp. 335–342. ISBN 0-20148-560-5.

[15] Boyd, S. K., Lenore, J., Lichti, D. D., Salkauskas, K., and Chapman, M. A., Oct. 1999, "Joint Surface Modelling with Thin-plate Splines," *Journal of Biomech. Eng.*, **121:5**, pp. 525–532.

[16] Ateshian, G. A., 1993, "A Least-squares B-spline Surface-fitting Method for Articular Surfaces of Diarthrodial Joints," *Journal of Biomechanical Engineering*, **115**, pp. 366–373.

[17] Ateshian, G. A., 1995, "Generating Trimmed B-spline Models of Articular Cartilage Layers from Unordered 3D Surface Data Points," *Proc. ASME Bioengineering Conference*, pp. 217–218.

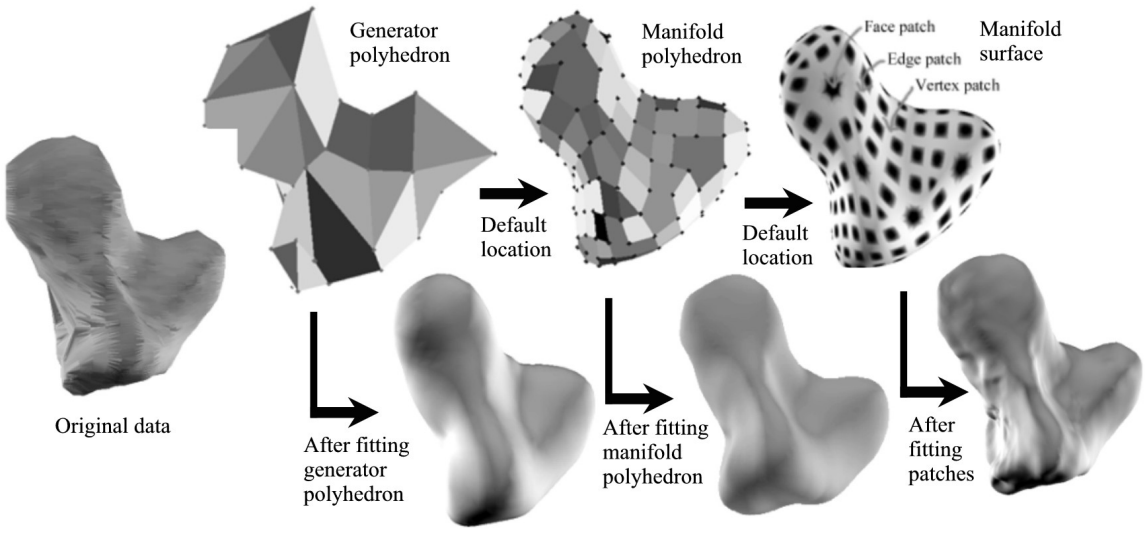

Figure 1: From left to right top: The original data, shown triangulated. The generator polyhedron, the manifold polyhedron, and the proto-type manifold surface, showing the local parameterization. Bottom: After fitting the generator polyhedron. After fitting the manifold polyhedron. After fitting the individual patches.

[18] Scherrer, P. K. and Hillberry, B., 1979, "Piece-wise Mathematical Represntation of Articular Surfaces," *J. Biomechanics*, **12**, pp. 301–311.

[19] Bartels, R., Beatty, J., and Barsky, B., 1987, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, first ed.

[20] NAG, 1993, *NAG Fortran Library*. 1400 Opus Place, Suite 200, Downers Grove, Illinois 60515: Numerical Algorithms Group.

[21] Fowler, B. and Bartels, R. H., July 1991, "Constraint Based Curve Manipulation," *Siggraph course notes 25*.

[22] Geiger, B., "Nuages Reconstruction Package," *INRIA*.

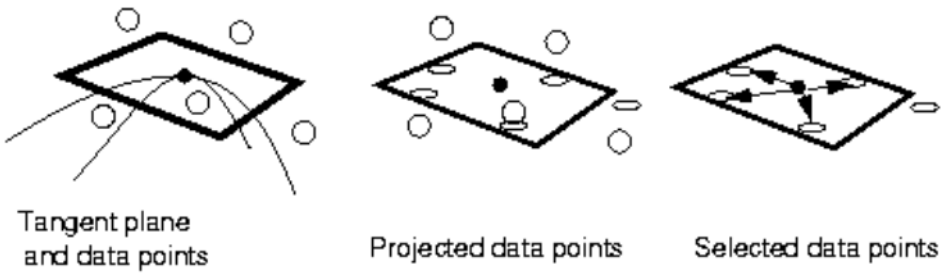

Figure 2: Finding the four data points to interpolate between for an additional patch constraint. Left: the surface point, the tangent plane, and the nearby data points. Middle: the projected data points. Right: the selected points, shown in the tangent plane.

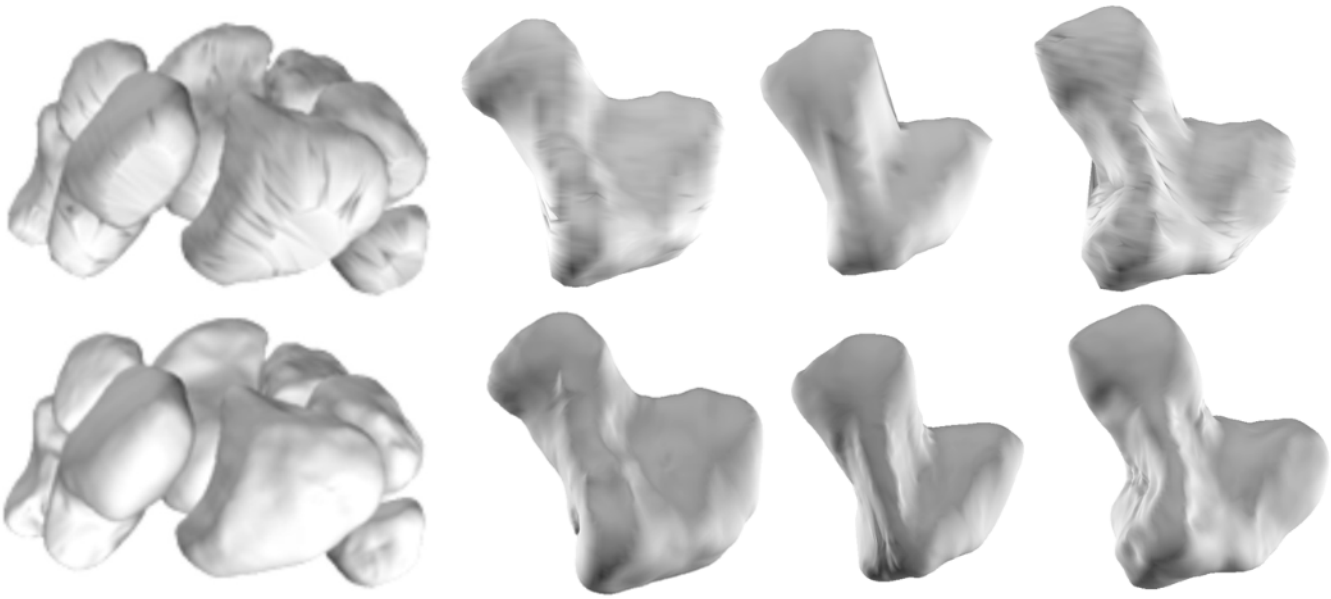

Figure 3: The carpal bones from one person and three hamates from different people. Top: Meshes produced from the data points using Nuages [22]. Bottom: Manifold surfaces. Note the striation in the Nuages meshes where the slicing planes become parallel to the surface.