Reverse Engineering Trimmed NURB Surfaces From Laser Scanned Data

Authors: Ben Steinberg[1], Anshuman Razdan[2], and Gerald Farin[3]

Abstract

A common reverse engineering problem is to convert several hundred thousand points collected from the surface of an object via a digitizing process, into a coherent geometric model that is easily transferred to a CAD software such as a solid modeler for either design improvement or manufacturing and analysis. These data are very dense and make data-set manipulation difficult and tedious. Many commercial solutions exist but involve time consuming interaction to go from points to surface meshes such as BSplines or NURBS (Non Uniform Rational BSplines). Our approach differs from current industry practice in that we produce a mesh with little or no interaction from the user. The user can produce degree 2 and higher BSpline surfaces and can choose the degree and number of segments as parameters to the system. The BSpline surface is both compact and curvature continuous. The former property reduces the large storage overhead, and the later implies a smooth can be created from noisy data. In addition, the nature of the BSpline allows one to easily and smoothly alter the surface, making re-engineering extremely feasible. The BSpline surface is created using the principle of higher orders least squares with smoothing functions at the edges. Both linear and cylindrical data sets are handled using an automated parameterization method. Also, because of the BSpline's continuous nature, a multiresolutional-triangulated mesh can quickly be produced. This last fact means that an STL file is simple to generate. STL files can also be easily used as input to the system.

Introduction

Reverse engineering, i.e. creating a digital representation of a physical object, is a powerful technique. As more of the engineering world goes digital i.e. vendors, suppliers and manufactures work from the same CAD model, it becomes even more important to convert the paper drawings into actual CAD model. Many times a part undergoes several design changes and after a few iterations it is difficult to predict how much the actual part has veered off from the original design intent. Also, many parts and objects which are still in high demand today, were created before CAD had become so mainstream. Having a digital representation of these objects makes upgrades and future analysis more efficient. Another reason is that for certain disciplines such as toy manufacturing the original character or model is created in clay by the artist. Its easier (more so because of the tradition) to sculpt the original rather than create in a CAD modeling software. In other cases even if the original part (such as a consumer part) is created in CAD, it may get modified (sanding etc.) as it undergoes

---

[1] Graduate Research Associate, PRISM, Arizona State University, Tempe AZ 85287-5106, USA.
[2] Technical Director PRISM, Arizona State University, Tempe AZ 85287-5106, USA. Email: razdan@asu.edu
[3] Professor, Dept. of Computer Science and Engineering, Arizona State University, Tempe AZ 85287-5106, USA

prototype feedback process. The only way to get the changes into the CAD model is to reverse engineer them. In addition, the need for automated inspection and verification of rapid prototyped and manufactured parts is also driving the field.

Part Digitizing and Reverse Engineering

Recently many accurate data acquisition systems have come to market. There are three main types of systems, the touch probes and Coordinate Measuring Machines (CMM), optic or laser based systems and volumetric or CAT and MRI systems. At present the PRISM lab has two Cyberware laser digitizers. In this paper we will focus on data collected via laser scanning techniques only, although the same process could be applied to data collected from other systems as well such as an STL file. Each system has different resolution and accuracy of measurement. The Cyberware scanner that was used has an accuracy of ±0.035mm. The problem of reverse engineering (from laser scanned data) is described as follows.

The part to be to be reverse-engineered is scanned with the laser digitizer. Based on the complexity of the part, it may take several scans to get all the geometry. The main reason for this is optical occlusion. If the laser or light cannot reach a portion of the object being scanned then no data is collected. Therefore, it requires more than one scan to capture the geometry. The multiple scans after cleaning (removal of supports and other artifacts) is transformed to a single coordinate system. Cyberware provides software to achieve the *zipping* process. Most systems provide one form or another to accomplish this task. Different digitizing systems do it differently but the end result is the same. The scanned data is a point cloud data with a triangulation. We use the triangulation to detect bounding and dangling edges only. The captured data can be taken directly into the CAD software, but it is just points and the triangulation. It is tedious to work with such a large number of points and requires tremendous amount of memory (RAM) on the computer. None of the commercial software we tried was up to the task. Mathematical representation of the same data with NURB (Non Uniform Rational BSpline) or BSpline surfaces (see Section III for a brief explanation of NURB Surfaces) is much more efficient both in memory requirement and ease of editing. This process is also called surface fitting. Once the equivalent surface is successfully created, it can then be imported into the CAD software. If the CAD modeling software has solid modeling capabilities, the surfaces can be joined together to get a single coherent solid model. Two issues dictate the way the surfaces are created. These are accuracy and ease of surface creation.

Previous Work

Many commercial software are now available that provide the user to interactively create surfaces from the digitized data. Of note are Imageware's Surfacer, Maya (Alias), Geomagic's Wrap, Nvision, etc. See [2] for a discussion of the hardware and software vendors. Hoppe et al.[6,7,8] and Lounsbery [10] also give methods to reduce the point cloud data and provide triangulation algorithms. Their work is motivated simply by the need for data reduction for faster display rather than geometric modeling requirements. The mulitresolution wavelet modeling is excellent for reducing the data to be displayed quickly, however, it does not serve well in a CAD setting since currently no solid modeler supports wavelet based models.

Current solutions (at the time of writing this paper) involve time consuming interaction to go from points to surface meshes. Typically the user is required to interactively select a region from the set of digitized points, or select a small region to create a network of curves. The curves then are used as the basis for skinning or lofting operation. Even for simple regions it requires a great deal of interactive input and several hours of work to create the network of curves. The quality of the surface is also dependent on the skill level of the user or the operator since the key points, such as regions of high curvature need to be captured in the network of curve for the resulting surface to be accurate.

Our approach differs from current industry practice in that we produce a mesh with little interaction from the user. The user can produce any degree BSpline surfaces and can choose the number of segments, tolerance etc. as parameters to the system. The BSpline surface is both compact and continuous. The result is an increase in automation and a decrease in needed user input. The combination of the least squares method, smoothing functions and automated data parameterization conversion, produces extremely accurate BSpline surface approximation. These surfaces are smooth, continuous and have well behaved boundaries. In addition we use trimming curves to preserve holes that part of the scanned object as well as define non-rectangular boundaries.

Methods

Brief definition of NURB Surfaces

A point on a NURB surface can be mathematically represented as:

$$X_{ij}(u, v) = \Sigma\Sigma\, w_{ij}\, d_{ij}\, N_i^m(u)\, N_j^n(v) \,/\, \Sigma\Sigma w_{ij} N_i^m(u)\, N_j^n(v) \qquad (1)$$

Where m and n are the degrees of the surface, d_{ij} are the control points of the surface, w_{ij} are the weights attached to control points. N_i^m and N_j^n are the BSpline basis function, which depend on the degree and parameterization of the surface. For more detail on basis functions and NURB Surfaces the reader is referred to [3,4,9]. There is also an important component called the knot sequence that influences the shape of the NURB. In the case of a surface, there is a knot sequence for both the u and v directions. The basis function evaluation depends on how the surface is parameterized, and the surface parameterization is based on the knot sequences.

Brief description of the Least Squares Method (LSM)

Given a set of points ***P*** in $\Re^3$, the goal is to find a surface $\boldsymbol{X(u,v)}$ such that the sum of Euclidean distances from all points in ***P*** to their corresponding locations on $\boldsymbol{X(u,v)}$, are minimized:

$$\text{Min } \Sigma \|P_i - \boldsymbol{X}(u_i,v_i)\|^2 \quad i=1..K \qquad (2)$$

where K is the number of points in the point cloud data. This guarantees the best-fit surface with minimum error for the given input parameters. The least squares method involves creating a standard equation:

$$\boldsymbol{A}_{KxR}\, \boldsymbol{x}_{Rx1} = \boldsymbol{b}_{Kx1} \qquad (3)$$

Where R is the number of control points $\boldsymbol{d}_{ij}$, $\boldsymbol{x}$ represents the unknown $\boldsymbol{d}_{ij}$, and $\boldsymbol{b}$ represents the set of data points $\boldsymbol{P}$. $\boldsymbol{A}$ is the matrix of basis function contributions for the data set $\boldsymbol{P}$. Applying $\boldsymbol{A}^T$ to both sides of Equation (3) results in:

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b} \qquad (4)$$

Equation (4) gives us the normal equations, a common form for the least squares method [1,9].

Our Approach

Scanning and Preprocessing

The object of interest is placed on the scanner. The scanning process, even at maximum resolution takes about 17 seconds per single pass. Depending on the complexity of the object's geometry, more than one scan may be necessary. Once the part is scanned, a pre-processing step is performed before surface fitting. The pre-processing identifies any boundary edges of the surface and any holes. At this time the user has a choice to fill the holes or keep them as trimmed holes. Since the boundary of the surface is rectangular i.e. we fit a surface that is larger than the data. Therefore, we have to identify the boundary trimming curve. The next step is to collect the necessary user input.

User Input

The user selects the resolution (degree, and number of BSpline segments) of the surface in the $\boldsymbol{u}$ and $\boldsymbol{v}$ directions. Objects which have a large number of convolutions generally need to have large amount of *segments* in both the $\boldsymbol{u}$ and $\boldsymbol{v}$ directions. Each *segment* represents a low degree polynomial surface, which is guaranteed to be at least C^r ($\boldsymbol{r}$ = degree-1) continuity with its neighboring segments. The more segments that a NURB surface has, the more *flexible* it is for fitting to a data set. The higher the degree of the NURB the greater the continuity between segments. Finally the user selects a parameterization scheme based on the topological nature of the scanned data. The two schemes that are currently implemented are functional and cylindrical.

Parameterization

Once the user has chosen the number of segments in the $\boldsymbol{u}$ and $\boldsymbol{v}$ directions, each data point in $\Re^3$ is mapped to a $\boldsymbol{u},\boldsymbol{v}$ parameter pair in the $\Re^2$ domain. This mapping is based on the user's choice of parameterization scheme. The functional scheme maps the points onto the XY plane, and then finds their distribution in the plane. Cylindrical mapping uses a calculated centroid for the object and maps the points around a cylinder and is used for objects that closed in one direction. Currently, the cylindrical mapping assumes the object revolves around the vertical-axis as it passes through the centroid, however it would be possible to calculate the parameters around an arbitrary axis. The user would specify this axis if a best guess of the long axis of the object were not satisfactory.

Smoothing Functions

At this point, the linear system is setup and the *A* matrix is complete. However, one of the problems with the least squares method is that *holes* or missing data can potentially cause the matrix to become singular. Each segment has a set of associated parameter values in ***u*** and *v*. If none of the data mapped into the domain fall within a given segment, the matrix becomes singular. If a segment is sparsely populated, it will behave poorly. We use smoothing functions to prevent poorly behaved regions from being generated. The smoothing functions add non-zero rows to the *A* matrix preventing the matrix from becoming singular [5]. The twist of a NURB segment is measure of how non-planar that quadrilateral segment is. Setting the twist to zero guarantees that the behavior of a segment in a region of little to no data is stable by forcing the segment to conform to the shape of neighboring segments. Since the splines are defined over a rectangular domain, any *rounded* shape data leaves segments unsupported around the border of the spline. The smoothing functions also help to force these border segments to continue in the direction of segments that do have data. The surface therefore continues smoothly past the areas of no data.

Trimming the Surface

While the outer region of the surface, which has no data associated with it, remains smooth, this region is still extraneous. This extra region is therefore trimmed away using a boundary-trimming curve. The trimming curve is automatically fit from the border points of the data set. While this is an automated process, we are still refining it. Currently, a very high number of curve segments are used to ensure a close fit around boundaries which oscillate. This may result in a trimming curve that undulates because of noisy boundary data. However, we are in the process of implementing alternative methods (See Conclusion). The removal of the extra surface leaves a relatively smooth edge which follows the shape of the actual object (see section Results). The surfaces are output as IGES trimmed NURB surfaces. These can then be read into any CAD solid modeling software which can read IGES surfaces. STL output can either be generated from the surfaces or from the resulting solid or surface model in the CAD software.

Results

We present some examples to show our results. The first example is of the trimmed hood and side fender of a toy car. The two pieces were individually scanned. Figure 1(a) and 2(a) show the raw digitized points. Figures 1 (b) and 2(b) shows the BSpline network with trimming curve. Figures 3 shows the trimmed surfaces in Maya (Alias) with a blend. The hood surface was a 30 x 30 segment bi-quadratic BSpline with a 71 segment cubic trimming curve. The side fender was modeled with a 30 x 20 bi-quadratic BSpline and a 71 segment cubic trimming curve. They were merged using a fillet blend function in Maya (Alias). Each model was fit to the original data with an average error of less than 0.09mm.

The second example is of a flashlight. The flashlight was scanned in two stages, the top and the side. Figures 4 (a) and 4 (c) show the raw digitized points. For this example, we used a periodic BSpline solution since no seam was desired in the final surface. The periodic BSpline was 50 x 20 segments of degree two. The BSpline for the top portion was a 5 x 5 bi-quadratic with a 15 segment cubic trimming curve. Figures 4 (b) and 4 (d) show the BSpline control networks and trim boundary curve. There is no boundary curve in Figure 4 (b) since it is a periodic surface. The two surfaces were again imported into Maya, stitched and converted into one coherent geometric surface model, Figure 5, using the same fillet blend function as in the previous example.

Conclusion

With very little interaction from the user our process can convert point cloud data from digitizers into IGES NURB surfaces, which can then be imported into a CAD solid modeler. Future work will focus on experiments with automated parameterization including non-uniform varieties. One reason that NURB surfaces provide powerful geometric models is due to the fact that the extra degrees of freedom the weight provides. Currently we set all the weights to one. Tighter fit or minimization of error can be achieved by manipulating the weights. The trimming curve has a tendency to undulate as the number of segments increase to ensure a tight fit around the boarder. We are planning to create the trimming curve in sections, joining the different pieces at junctures where the boundary data is naturally C^0. The trimming curves will be fit to small segment of data, and should be smoother than a single curve with a large number of segments.

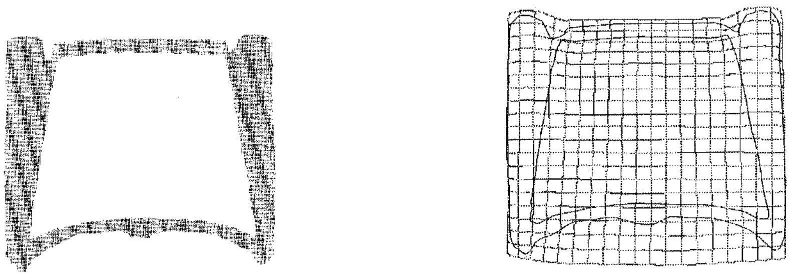

Figure 1: (a) Car top, raw points, (b) BSpline

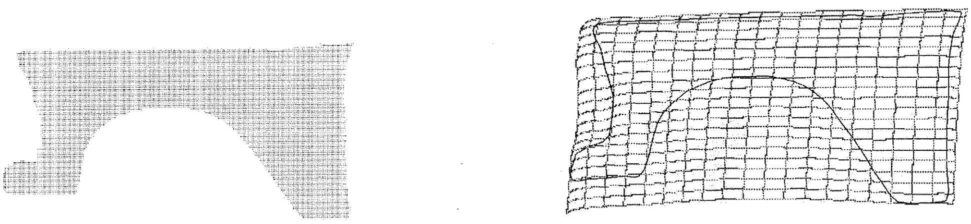

Figure 2: (a) Car side, raw points, (b) BSpline

Figure 3: Finished product. (a) Front view and (b) Side view

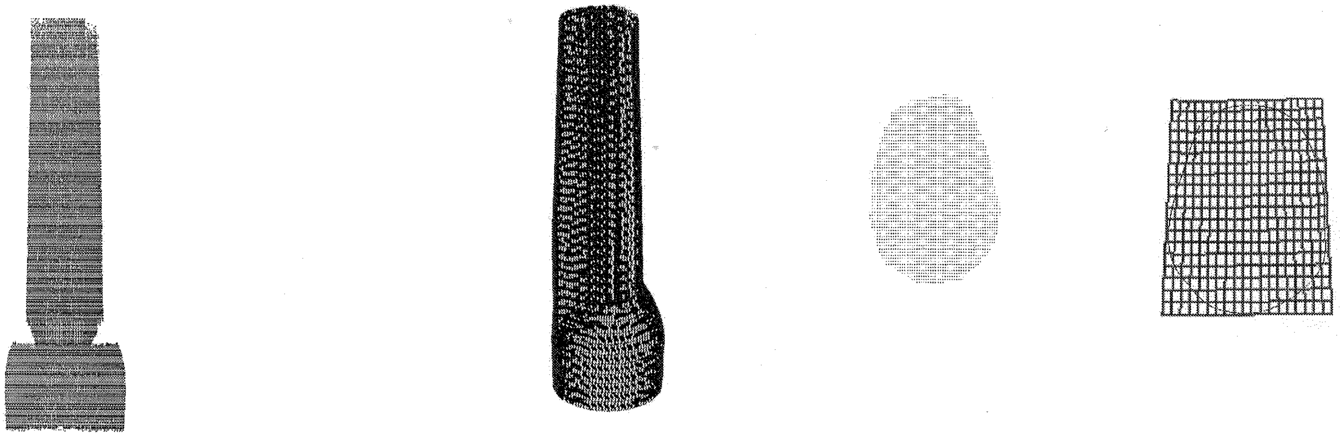

Figure 4: (a) Flashlight body, raw points; (b) Flashlight body, spline; (c) Flashlight top, raw points; (d) Flashlight top, spline

Figure 5: Finished product, two views

Acknowledgements

The authors would like to acknowledge D. Hansford for her invaluable ideas and suggestions. We would also like thank C. Gosser and G. Kattethota.

References

1. Å. Björk. Numerical Methods for Least Squares Problems. SIAM. Philadelphia, PA. 1996.

2. R. Broacha and M. Young. Getting from points to products. *Computer-Aided Engineering*, 1995.

3. G. Farin. Computer Aided Geometric Design. Academic Press. New York, NY. 1997.

4. G. Farin. NURB Curves and Surfaces, from Projective Geometry to Practical Use. AK Peters. Wellesley, MA 1996.

5. G. Farin and D. Hansford. Shape from permanence. Submitted, CAGD.

6. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. Surface reconstruction from unorganized points. In *Computer Graphics, SIGGRAPH '92*, volume 26, 1992.

7. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. Mesh optimization. In *Computer Graphics, SIGGRAPH '93*, volume 27, 1993.

8. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. Piecewise smppth surface reconstruction. In *Computer Graphcis, SIGGRAPH '94*, volume 28, 1994.

9. J. Hoschek and D. Lasser. Fundamentals of Compuoter Aided Geometric Design. AK Peters, Wellesley, MA. 1989.

10. M. Lounsberry, S. Mann and T. DeRose. Parametric surface interpolation. *IEEE Computer Graphics and Applications*, 1992.