Smoothing Space Curves with the MLS Projection

Lavanya Sita Tekumalla[1], Elaine Cohen[1]
[1] School of Computing, University of Utah

Abstract

Smoothing space curves has several applications in reverse engineering, CAD modeling and animation. We propose a 3D curve smoothing algorithm with the primary focus on smoothing boundaries of point clouds obtained during reverse engineering laser scanned models.

While several point cloud denoising methods exist that handle the normal noise in the data, the boundary curve may still contain tangential noise. Our space curve smoothing algorithm can be used to obtain models with smooth boundaries.

keywords: denoising, boundary smoothing, MLS projection, point clouds

1 Introduction

With the advent of high precision 3D scanners, the process of reverse engineering objects by scanning models is gaining importance in the field of design and manufacturing. The data produced as a result of scanning is often noisy with outliers. While several methods exist, to smooth noisy scanned data, most of them focus on smoothing the normal noise in the surface. However, the boundary of the surface might have tangential noise that is not smoothed. In order to obtain a smooth surface for use in the process of manufacture and CAD applications it is essential to smooth the boundary curve separately.

Apart from this, 3D curve smoothing is a well studied problem and has several other applications in 3D surface registration and motion capture. Space curves play an important role in computer vision and image registration where feature curves need to be matched. Noise in space curves often poses difficulties [1]. Also, positional data that corresponds to the path followed during the motion of a body is often a space curve. Such data, when measured, might result in noisy space curves that need smoothing for the purpose of analysis and for other applications such as motion smoothing in virtual reality [2].

We wish to extend the MLS projection operator for surfaces, that projects each point near the surface onto the surface, proposed by [3], to curves, for smoothing space curves.

2 Previous Work

Previous approaches on curve smoothing involved approximating the data with a single parametric surface by minimizing the L^2 norm. B-spline curves and Bezier curves have been the representations of choice for such applications. In [1], a sequence of points are approximated using a B-spline for smoothing curves for feature matching.

Another way to deal with noisy scattered data is to minimize a combination of the L^2 norm and the smoothing norm (refer to [4]). The functional that is minimized has two terms. One measures the deviation of the data points from the fit, and the other measures the non-smoothness of the fit. However, these methods involve finding a global parameterization for all the points. However, when we have unorganized data, parameterizing data is a non-trivial task.

Considerable work has been done to smooth triangular meshes (refer to [5]), and to denoise point clouds for reverse engineering [6, 7, 8, 9]. In [7], the MLS projection is used to smooth point set surfaces. In [9], a modified MLS projection operator is used, that is more robust for the purpose of smoothing. Our method is inspired by the MLS projection procedure for surfaces and aims to smooth boundaries of surfaces, that are space curves, to remove tangential noise using a similar procedure.

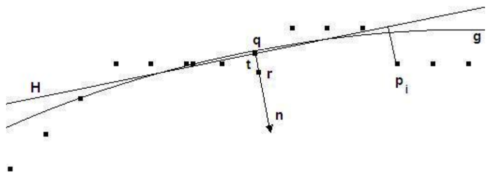


Figure 1: MLS Projection procedure for surfaces

3 A Review of the MLS Projection for Surfaces

The MLS projection procedure was proposed by [3] and [7] to deal with meshless surfaces. Given a point set, the MLS projection operator projects a point r near the surface onto a unique manifold surface implicitly defined by the set of points. This surface can be defined as the set of fixed points of this operator, that project onto themselves. Another desirable property of this is that the resulting manifold surface is guaranteed to be infinitely smooth, if the weighting function in the moving least squares process is infinitely smooth.

The MLS projection operator proceeds in two steps. To project a point r, the first step requires finding an optimal local reference plane for the neighborhood of r by minimizing the L^2 norm of the weighted perpendicular distance of points p_i in the neighborhood from the optimal reference plane. If n is the normal to the plane and t the distance of the plane from r (figure 1), $\sum_{i=1}^{N} \langle n, p_i - r - tn\rangle^2 \theta(||p_i - r - tn||)$ is minimized with respect to n and t, where θ is a gaussian weighting function defined as $\theta(x) = e^{(-x^2/h)}$, h controlling the standard deviation. This is a non-linear minimization process. A local parameterization is obtained by projecting each point in the neighborhood onto this reference plane. The next step involves fitting a local bi-quadratic polynomial surface g using the moving least squares technique. That is, we find a g to minimize $\sum_{i=1}^{N} (g(x_i, y_i) - f_i)^2\theta(||p_i - q||)$ where $q = r + tn$ (q is the projection on the best fit plane), (x_i, y_i) are the parameter values of p_i in the local reference plane and $f_i =< p_i - q, n >$ is orthogonal to the local reference plane.This polynomial when evaluated at the point q, gives the desired MLS projection.

4 MLS Projection Procedure for Space Curves

In section 3 the MLS projection for surfaces, based on the pioneering work of Levin [3] has been described. This section aims at extending this procedure to space curves.

Our method for MLS projection for curves is similar to that for surfaces and has the following steps. For every point r on the curve,

1. Find a local reference line : Find a local neighborhood N_r consisting of N points. Suppose u is a unit vector in the direction of the optimal reference line and q is the projection of r on u (as shown in figure 2), for every point P_i belonging to N_r, $\sum_{i=1}^{N} ||(p_i - q) - \langle p_i - q, u\rangle\, u||^2\theta(||p_i - q||)$ is minimized with respect to q and u where θ is the Gaussian weighting function defined as $\theta(x) = e^{(-x^2/h)}$. This is a non-linear minimization process. Representing u using spherical coordinates we have $u = (\cos\gamma\cos\phi, \cos\gamma\sin\phi, \sin\gamma)$ for some γ and ϕ. This gives us two degrees of freedom for γ and ϕ. Making use of the fact that the direction $q - u$ is perpendicular to u, we have two degrees of freedom for the point q. Hence we have four degrees of freedom during the minimization. We use the Powell minimization method to get the optimal values for q and u.

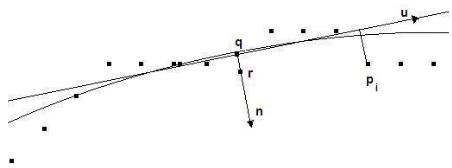


Figure 2: MLS Projection procedure for space curves

2. Now a local orthonormal basis, u, $(r - q)$ and $(u \times (r - q))$ is formed. The neighborhood is transformed to this local orthonormal basis with q as the origin. For every point p_i, the new point in the local coordinate system become $q_i = (< (p_i - q), u >, < (p_i - q), (r - q) >, < (p_i - q), (u \times (r - q)) >)$
3. Finally a quadratic curve is fit and the point

is projected onto the curve. Since the curve might not be planar, we use a parametric quadratic curve $g(t) = (t, v(t), w(t))$.

4. The curve evaluated at $t = 0$ using the moving least squares method gives the desired MLS projection.

The value of h, that determines the standard deviation of the weighting function θ, plays an important role in this process. Figure 5 shows different curves that result when smoothing is performed with different h. A higher value of h tries to make the curve smoother while a lower value attempts to adhere closer to the input.

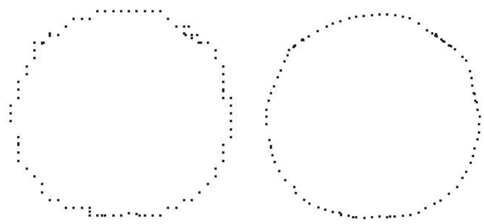
This smoothing method, when applied by itself, suffers from the fact that it cannot preserve sharp features in the input. In the context of smoothing boundaries, the boundary curve might contain sharp features, though the interior of the surface might not. For instance, if we have a rectangular patch of an object segmented out, the corners of the rectangle are smoothed when the above method is used.To avoid this problem, we use user intervention, to point out the sharp corners in the boundary. Now the boundary smoothing process is carried out for each individual piece of the boundary to preserve sharp features.

Also, this method might not preserve the order of the points when an ordered point set is given. This is because, when the points are parameterized locally by projection, the outliers may not be parameterized in the right order. However, this is not a major concern when dealing with point clouds.

5 Results

Figures 3 and 4 show curves smoothed by this process. In figure 5 the technique is illustrated for a non-planar space curve. We have smoothed this curve with different values of standard deviation (h) and the results are shown in the figure. As seen in the figure, using different values of standard deviation produces different level of smoothing.

Figure 6 show the technique applied to smooth the boundaries of point clouds. The noisy boundary curves are shown in figures 3 and 4. In the absence of boundary information, an underlying triangulation may be used to obtain the boundary. Several well known triangulation algorithms exist for this purpose in the absence of connectivity information.

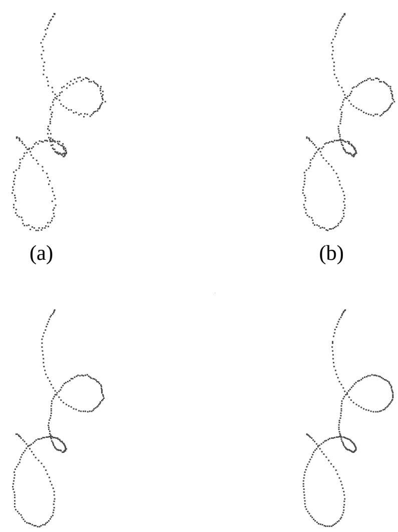


(a) (b)

Figure 3: A noisy circular curve smoothed using our method

(a) (b)

Figure 4: A noisy curve, smoothed using our method

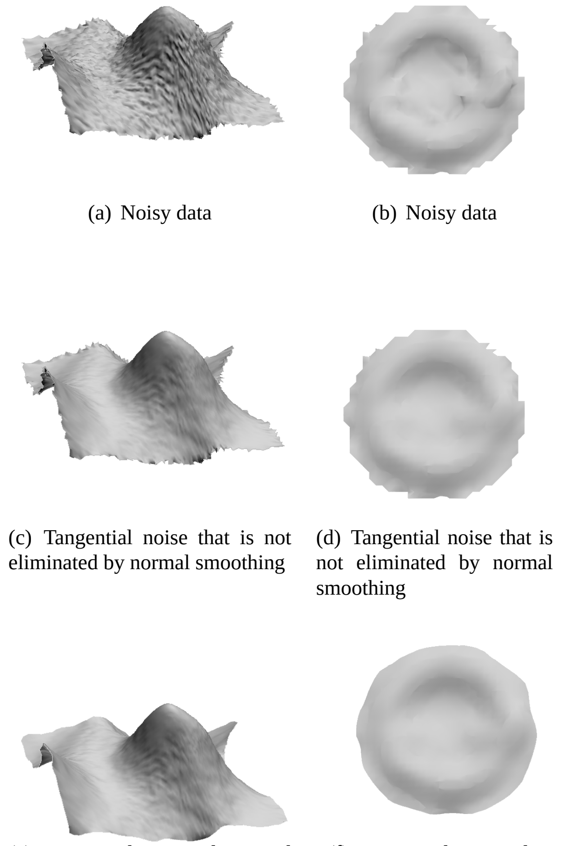


(a) (b) (c) (d)

Figure 5: In figure (a), a noisy space curve is shown. The subsequent figures show smoothed curves with increasing value of standard deviation (of 2.5, 4.5 and 6.5 times the average point spacing respectively), that demonstrates the increasing smoothing effect

6 Conclusion

We have provided a 3D curve smoothing algorithm based on the MLS projection that can be used in



(a) Noisy data (b) Noisy data

(c) Tangential noise that is not eliminated by normal smoothing (d) Tangential noise that is not eliminated by normal smoothing

(e) Tangential noise eliminated by our curve smoothing method (f) Tangential noise eliminated by our curve smoothing method

Figure 6: The figure illustrates the boundary noise after normal smoothing and the results obtained when smoothing the tangential noise in the boundary using our method

a variety of applications. Our algorithm is simple, does not need an ordering of the input points and can handle sharp corners in the input.

7 Acknowledgement

This work was supported in part by ARO (DAAD19-01-1-0013). All opinions,findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.

References

[1] A. Guziec and N. Ayache, “Smoothing and matching of 3-d space curves,” *The International Journal of Computer Vision*, vol. 12, no. 1, pp. 79–104, January, 1994.

[2] M. W. M. K. S. S. C.C. Hsieh, Y.C. Feng and T. Woo, “Noise smoothing for vr equipment in quaternions,” *IIE (Institute of Industrial Engineering) Transactions*, vol. 30, pp. 581–587, 1998.

[3] D. Levin, “Mesh-independent surface interpolation,” *Geometric Modeling for Scientific Visualization*, pp. 37–49, 2003.

[4] P. Dierckx, *Curve and Surface Fitting with Splines*. Oxford Science Publications.

[5] A. Belyaev and Y. O. I.-K. Tel-Aviv, “A comparison of mesh smoothing methods,” *Bi-National Conference on Geometric Modeling and Computer Graphics*, pp. 83–87, February, 2003.

[6] M. Pauly and M. Gross, “Spectral processing of point-sampled geometry,” *Proceedings of ACM SIGGRAPH*, 2001.

[7] A. Marc, B. Johannes, C.-O. Daniel, S. Fleishman, D. Levin, and T. S. Claudio, “Computing and rendering point set surfaces,” *IEEE Transactions on Computer Graphics and Visualization*, vol. 9, pp. 3–15, 2003.

[8] C. Lange and K. Polthier, “Anisotropic fairing of point sets,” *Special Issue of CAGD (to appear)*, 2004.

[9] L. V. Boris Mederos and L. H. de Figueiredo, “Robust smoothing of noisy point clouds,” *SIAM Conference on Geometric Design and Computing, Seattle 2003*, 2004.