Mesh Generation and Refinement of Polygonal Data Sets

V. Savchenko
Hosei University
vsavchen@k.hosei.ac.jp
N. Kojekine
Tokyo Institute of Technology
karlson@stu.mech.titech.ac.jp
M. Savchenko
Japan Science and Tech. Corp.(JST)
maria@mech.titech.ac.jp

I. Semenova
Tokyo Institute of Technology
semenova@stu.mech.titech.ac.jp
O. Egorova
Tokyo Institute of Technology
egorova@stu.mech.titech.ac.jp
I. Hagiwara
Tokyo Institute of Technology
hagiwara@mech.titech.ac.jp

Abstract

This paper presents work in progress and continues a project devoted to developing shape modeling system based on implementation of radial based function (RBF) technology. In this paper, we study the opportunities offered by this technology to computer-aided design and computer graphics communities by looking at the problems of surface generation and enhancement. Experimental results are included to demonstrate the functionality of our mesh-modeling tool.

Keywords: radial basis functions, mesh generation, simplification

1. Introduction

Surface reconstruction and remeshing have become very important today for computer-aided design (CAD) and computer graphics (CG), and Cyber World technologies. For instance, the quality of texture mapping depends on the shape of triangular elements. These questions are also very important for technologies related to engineering applications. Construction of a geometric mesh from a given surface triangulation has been discussed in many papers (see [1] and references therein). The whole process involves constructing a geometric mesh which is then optimized so as to improve the element's shape quality.

Point sets obtained by means of computer vision techniques are often non-uniform and even contain large missing areas of points. Reducing the complexity of such data is a challenging goal. Many algorithms for conversion and modification of such data are computationally expensive. The final goal of our project is to allow mesh to be generated from scattered point sets for finite element analysis. Various automatic mesh generation tools are widely used for finite element analysis. However, all of these tools may create distorted or ill-shaped elements, which can lead to inaccurate and unstable approximation. Thus, improvement of mesh quality is an almost obligatory step for preprocessing of mesh data in finite element analysis. Recently, sampled point clouds have received much attention in the CG community for visualization purposes (see [2,3]) and CAD applications (see [4,5]). Their information is processed by surface reconstruction algorithms and subsequently simplified and denoised. In spite of a flurry of activity in the fields of scattered data reconstruction, interpolation, and mesh modification, this matter remains a difficult and computationally expensive problem.

A vast amount of literature has been devoted to the subject of scattered data interpolation methods (see, for instance, [6,7]). Our work is primarily focused on the efficiency of implementation of radial basis functions (RBFs). This paper presents work in progress, and continues a project devoted to developing a system for shape modeling based on implementation of RBF technology. We report our preliminary results, and study the opportunities offered by this technology for CAD and CG communities by looking at the problems of surface generation and enhancement, which include polygon generation from unorganized points and shape smoothing, simplification, and improvement of mesh quality parameters of 3D polygonal sets.

Surface reconstruction methods can be broadly classified into global and local approaches. We show the applicability of compactly supported radial basis functions (CSRBFs) for local surface reconstruction by using a slightly modified Shepard method [8] for the case of restoration of elevation data. A space-mapping technique (mapping $\mathbf{R}^3$ to $\mathbf{R}^3$) based on RBFs is a powerful tool, which offers simple and quite general modification of simulated shapes. Thus the second goal is to apply a space-mapping technique based on RBFs to automatically defined portions of the input data to be simplified, and the third goal is to realize a local mesh enhancement based on the statistical characteristics of an initial triangle mesh.

Main contribution of the paper is a surface simplification method which uses the fact that RBFs offer a mechanism for obtaining extrapolated points of a surface. We present an approach for obtaining a realistic time response for sufficiently complex models (70K triangles). This approach is based on the idea of obtaining 3D coordinates according to a bending energy.

An analysis of existing methods has defined a central concept of our project: we consider that local surface reconstruction performed on point sets, triangle mesh extraction, and mesh simplification with simultaneous mesh improvement are integral parts of the project.

The rest of the paper is organized as follows. The next section gives an overview of papers related to this work. An algorithm for recovering surfaces by implementing the partition of unity is presented in Sec. 3. We discuss the simplification algorithm in Sec. 4, Sec. 5 presents experimental results, and sec. 6 contains concluding remarks.

2. Related works

In the last few years, surface reconstruction, smoothing, and remeshing have been studied intensively and several effective numerical algorithms have been reported. One approach is to use methods of scattered data interpolation based on minimum-energy properties (see [9,10,11]). The benefits of modeling with the help of RBFs have been recognized in many studies. To the best of our knowledge, the first publications on the use of discrete 2D landmark points were those of Bookstein [12]. RBFs were adapted for surface reconstruction in [13,14]. Methods of reconstructing a model on the basis of global reconstruction by using RBFs and CSRBFs produce sufficiently good approximations of a surface, but they suffer from two drawbacks: they take a great deal of time, and artifacts or “ghost” objects can appear as a result of the extraction of a surface from implicitly defined functions (see recent work such as [15]). To overcome the necessity of using normals to the surface and avoid extraction of an implicitly defined isosurface, one possible way, which is considered in our project, is to implement a local approximation based on the Shepard method [8], the so called partition of unity. Here we consider a slight modification of the Shepard method using CSRBFs as support functions, and information about the surface curvature.

Complex and detailed models can be generated by 3D scanners, and such models have found a wide range of applications in CG and CAD, particularly in reverse engineering. Nevertheless, it is useful to have various simpler versions of original complex models according to the requirements of applications. Recently, a tremendous number of very sophisticated algorithms have been invented to obtain a simplified model. One exceedingly good overview [16] presents a problem statement and a survey of polygonal simplification methods and approaches. Most existing simplification algorithms use such topological operations as vertex decimation, edge decimation, and triangle decimation. Vertex decimation methods remove a vertex according to a decimation cost defined by an error metric. In [17], an extremely fast simplification method based on a probabilistic optimization technique was proposed.

In different applications, especially in finite element analysis, the quality of the surface polygonization is also important (see, for instance, [1]). Two main ways of improving mesh quality exist. One is so-called clean-up, which modifies mesh topology by inserting or deleting nodes, or by local reconnection. However, sometimes it is necessary to minimize changes in the topology of a surface. Therefore, there are methods that improve mesh quality without any topological changes. The approach is called smoothing. Many smoothing techniques have been developed. Among the earliest methods are Laplacian smoothing [18] and its variations. In recent years, the CG community has paid more attention to mesh smoothing based on a signal processing approach, pioneered by Taubin in 1995 [19].

According to our experience, when the original model is simplified to less than 70 percent of its original complexity (see Figure 1), there is almost no visual difference between the original and simplified meshes. The simplification methods preserve the original shape quite well. Nevertheless, our implementation of the edge contraction method uses the idea of finding an optimal position based on minimization of a bending energy to place a new vertex. As our experiments show, such a scheme does not cost a great deal of time, and allows a rather reasonable simplification ratio of 90% reduction of the number of original points to be obtained.

3. Generation of Polygonal Data Sets

Recently, Wendland [20] constructed a new class of positive definite functions for 1D, 3D, and 5D spaces of the form

$$\phi(\mathrm{r}) = \begin{cases} \psi(r), & 0 \le r \le 1 \\ 0, & r > 1 \end{cases},$$

where $\psi(r)$ is a univariate polynomial whose radius of support is equal to 1. Scaling of the function $\psi(r/\alpha)$ allows any desired radius of support α. Nevertheless, even the use of CSRBFs does not provide a reasonable processing time for rather moderately sized point sets. Finite element methods (FEMs) are applied for restoration of scattered data, but they also have various drawbacks. From our point of view, methods based on the idea of local reconstruction are promising in CAD and CG applications dealing with huge amounts of scattered data.

The partition of unity method (PUM) for the construction of interpolation and approximation was pioneered by Shepard [8] and was later extended by Franke and Nielson [6]. In recent years, it has received much attention due to the works of Melenk and Babuska [21] and Krysl et al. [22].

Shepard's approximation on a set of scattered points $\boldsymbol{x}$ of domain Ω is as follows:

$$u_h(\boldsymbol{x}) = \sum_{I=1}^{N} \omega_I(\boldsymbol{x})\, u_I \ ,$$

where u_I are the nodal parameters, and $\omega_I(\boldsymbol{x})$ are the basis functions of compact support. They are constructed from weight functions $W_I(\boldsymbol{x})$ by means of the formula

$$\omega_I(\boldsymbol{x}) = W_I(\boldsymbol{x}) / \sum_{k=1}^{N} W_k(\boldsymbol{x}).$$

In this work, we have used the CSRB function

$$W_I(\boldsymbol{x}) = \begin{cases} (1-r)^4(4r+1), & 0 \le r \le 1 \\ 0, & r > 1 \end{cases},$$

where r = $||\boldsymbol{x} - \boldsymbol{x}_I||$ is the Euclidean distance between an interpolated point and an input point, and N is the number of points in a predefined area. Other choices of the weight function are also acceptable; however, theoretical proofs can be given to show that, to achieve extrapolation efficiency, weight functions, with small third derivatives should be used.

A general cover construction algorithm or partition of the domain Ω into overlapping rectangular patches ω_I to cover the complete domain has to be used. Let us note that our main premise is to take account of a surface curvature that might be useful for correct choice of a radius of support (r-sphere) for reconstruction taking into consideration the orientations of local surface elements. In our work, we have investigated rather a simple scheme taking account of the local geometry of a surface.

In our software implementation, we employ a standard approach for creating a binary tree from an initial point data set with an additional required parametric value K, which denotes the maximum number of points in a leaf. Once we have calculated the tree, we use this tree to search for the neighbors of any given point. Hoppe et al. [4] have demonstrated that eigenvalues λ_1, λ_2, and λ_3 of the covariance matrix of neighboring points could be used to produce normal estimates. These values λ_1, λ_2, and λ_3 also describe the surface variation σ and serve as an analog of a surface curvature. They also define the orientation of an ellipsoid containing the data. If the surface variation σ is larger then 0.3 – that is, if the data demonstrate a strong deviation from an average plane – we attract data from additional neighboring cells in accordance with the ellipsoid's orientation and increase the r-sphere. Figure 4(a) shows the surface variation σ of the elevation data.

4. Simplification Algorithm

Metrics based on object geometry properties are mainly used in simplification algorithms. Current simplification algorithms might be made more effective if an algorithm could produce an approximation of the simplified surface. In fact, we present an attempt to combine a simplification process with a surface approximation based on the use of RBFs.

Here, we shall give a short account of the shape transformation method used in the applications considered in this paper (for further references, see [23]). A space mapping in $\mathbf{R}^n$ defines a relationship between each pair of points in the original and deformed objects. For an arbitrary three-dimensional area Ω, the solution of the problem is well known: the volume spline $f(P)$ having values h_i at N points P_i is the function

$$f(P) = \sum_{j=1}^{N} \gamma_j \phi(|P - P_j|) + p(P), \qquad (1)$$

where $p = v_0 + v_1 x + v_2 y + v_3 z$ is a degree one polynomial. To solve for the weights γ_j we have to satisfy the constraints h_i by substituting the right part of equation (1), which gives

$$h_i = \sum_{j=1}^{N} \gamma_j \phi(|P_i - P_j|) + p(P_i), \qquad (2)$$

Solving for the weights γ_j and v_0, v_1, v_2, v_3 it follows that in the most common case there is a doubly bordered matrix, which consists of three blocks, square sub-matrices $\boldsymbol{A}$ and $\boldsymbol{D}$ of size $N \times N$ and 4×4 respectively, and $\boldsymbol{B}$**,** which is not necessarily square and has the size $N \times 4$.

We employed an approach that uses displacements of N control points as the difference between the initial and final geometric forms. Interpolation of (x,y,z) points is implemented in $\mathbf{R}^3$ and defines a relationship between the coordinates of points in the original and deformed objects. The inverse mapping function that interpolates the z-heights and is needed to calculate destination points z^d is given in the form

$$\boldsymbol{z}^s = f(P) + z^d, \qquad (3)$$

where the components of the volume spline $f(P)$ interpolating displacements of starting points $\boldsymbol{z}^s$ are used to calculate points to be processed.

Finding the optimal decimation sequence is a complex problem. The traditional strategy is to find a solution that is close to optimal; this is a greedy strategy, which involves finding the best choice among all candidates. Our simplification algorithm is based on an iterative procedure for performing simplification operations. In each iteration step, candidate points for an edge collapse are defined according to a local decimation cost of points belonging to a shaped polygon. After all candidates have been selected, we produce a contraction step by choosing an optimal point.

In our implementation, we used and compared the computation of local decimation costs in two ways. The simplest way is to compute a decimation cost as a minimal distance between the central point of shaped polygon and those of its neighbors. Another way is to apply a specific error metric. We propose using the bending energy $\boldsymbol{h}^t\boldsymbol{A}^{-1}\boldsymbol{h}$ as an error/quality cost to select candidates for an edge collapse. The central point of a shaped polygon is considered as a point that can slide to the neighboring points. The selection of candidate points is made according to the bending energy. The degree of contraction operation is estimated by minimizing the bending energy. We exploit a simple idea that the more smoothly we transform a central point, the fewer residuals there will be between an initial mesh and the subsequent mesh. In this step, we form a list of points to be contracted; this list contains a number of candidate points. In the contraction step we eliminate processing of points that can be contracted twice or more.

Vertex placement is produced in two steps. In the first step we generate an optimal position on the line connecting two vertices of an edge to be contracted. Our main premise here is to use the bending energy to obtain the “best approximated surface”. Recall that the spline (2) (determined by the set of variable control points P_i that belong a union of two shaped polygons) provides a minimization of the bending energy $\boldsymbol{h}^t\boldsymbol{A}^{-1}\boldsymbol{h}$, where $\boldsymbol{h}_i$ are so-called heights, for the case in which space transformations $\boldsymbol{h}_i = \boldsymbol{Y}^0_i - \boldsymbol{X}^0_i$ are the initial and final points. To find an optimal point we use an approach of spline relaxation along line elements, pioneered by Bookstein for the 2D case [12]. It allows the spline method to be extended so that some of the target 3D landmarks are free to slide along lines; in our case, two vertices belonging the edge collapse slide in 3D from their nominal positions $\boldsymbol{Y}^0$ along the directions $u_j = (u_{jx}, u_{jy}, u_{jz})$ over the lines $\boldsymbol{Y}^0{}_{ij} + t_j u_j$; $t_j \in \boldsymbol{T}$ (parameter vector). Let us note that neighboring points are not free to slide.

The solution to this problem is achieved for the parameter vector $\boldsymbol{T}$ over which the energy of the corresponding spline is minimized:

$$\boldsymbol{T} = -(\boldsymbol{U}^t \boldsymbol{L}^{-1} \boldsymbol{U})^{-1} \boldsymbol{U}^t \ \boldsymbol{L}^{-1} \boldsymbol{Y}^0,$$

where the matrix $\boldsymbol{L}$ is a block-diagonal matrix consisting of three $\boldsymbol{A}$ matrices. When the parameter vector $\boldsymbol{T}$ is defined, an optimal point belonging to the line connecting two contracted vertices is calculated. In fact, the optimal point is generated at the next step according to the following scheme:

• Define a tangent or an average plane passing through the point calculated in the previous step. Once we have information about the neighboring points, which can be achieved by estimating the tangent plane, we can estimate the local surface properties. The values λ_1, λ_2, and λ_3 describe the surface variation σ and can serve as an analog of a surface curvature that is used for generation of polygonal data sets to make a decision about the complexity of the local reconstruction area. Assuming that λ_1 is minimal, λ_1 describes the variation along the surface normal, and the directions corresponding to the eigenvalues λ_2, λ_3 define a tangent plane; we proceed as follows:

• Apply a rotation to the neighboring points so that the nearest plane is perpendicular to the *z*-direction.

• Calculate the differences or deviations for local *z*-directions that define the inverse mapping function (3).

• Apply a transformation for the selected point.

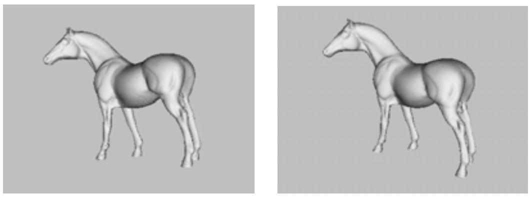
• Produce an inverse rotation of the calculated point.

Let us note that the surface variation σ can be used as an estimation of a local curvature of current mesh to distinguish the areas having different geometric characteristics for preserving object features. Afterwards, when the original model has been simplified, we repeat the above steps to produce new updates.

5. Examples

All the examples in this paper were run on our test configuration: AMD Athlon 1000 MHz, 128 MBRAM, Microsoft® Windows 2000, and ATI Radeon 8500 LE.

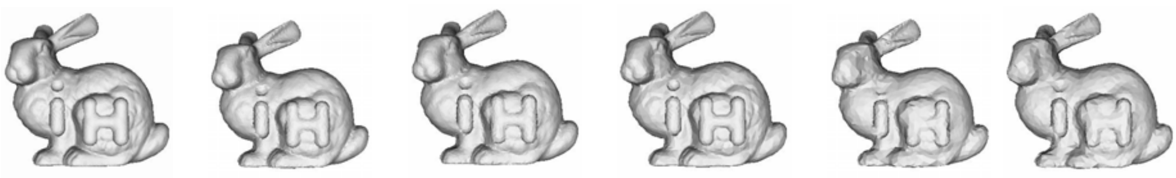
We demonstrate our work on four models. In Figure 1 we show that the algorithm exhibits very good features when a model is simplified to 30% of its original complexity. Figure 2 shows examples of polygon simplification of a modified "Stanford Bunny" to demonstrate the preservation of sharp features of the initial model.



(a) (b)

Figure 1. (a) Original "horse" model (96966 triangles), (b) Simplified model produced accordingly to our approach (30% of the original number of triangles).

We compare our approach with the software algorithm from [24]. As can be observed in Figure 2, our simplification algorithm does not provide such a good quality of simplification as the algorithm given in [24]. However, it has the advantages that the processing time is almost three times as fast, and no user-tuned parameters are required. At the same time, it guarantees overall smoothness and preservation of features. The results of time estimation are given for non-optimized code.



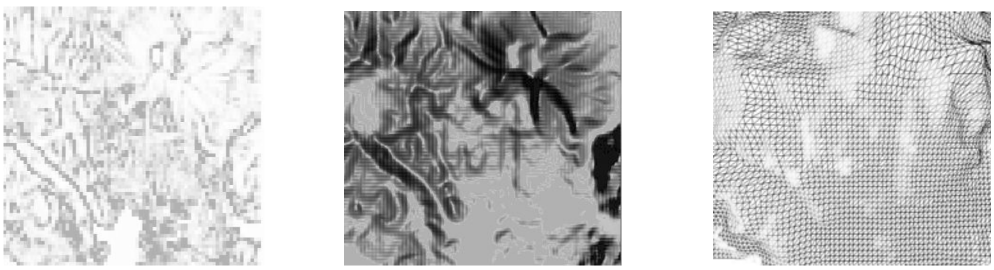
(a) (b) (c) (d) (e) (f)

Figure 2. (a) Modified "Stanford Bunny" model, simplified according to the algorithm [24] (30% of original data, processing time: 158.989 sec). (b) Simplified model (30%) using a simple geometric error metric. (c) Simplified model according to our approach (30%, processing time: 59.737 sec). (d) Model simplified according to the algorithm [24] (10% of original data, processing time: 246.124 sec). (e) Simplified model (10%) using a simple geometric error cost. (f) Model simplified according to our approach (10%, processing time: 70.521 sec).

In fact, PUM does not provide overall smoothness, nevertheless, it can be improved by applying RBF smoothing. Let us note that we process points nearest to the point that should be moved, according to the scheme defined for placement of an optimal point, as was discussed in Section 4. Figure 3 shows the effect of RBF smoothing of polygonal surfaces. To define the inverse mapping function, a different number of vertices N can be used for interpolation. The larger the number N we select, the more features will be preserved by the smoothing. To achieve a better smoothing effect, several iterations of the smoothing algorithm could be applied. Figure 4 shows the results of reconstruction using the approach discussed in section 3.

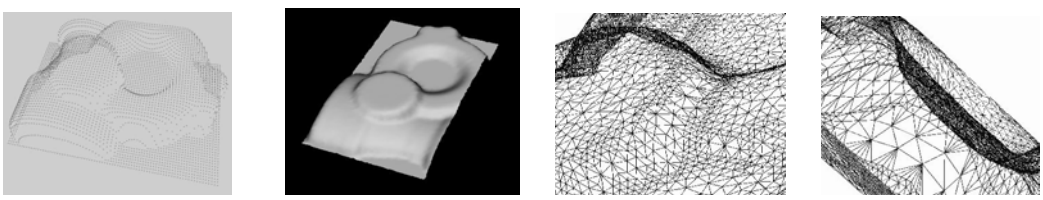
(a) (b)

Figure 3. (a) The original “Stanford Bunny” model (69451 polygons). (b) Smoothed model after one iteration based on 11-point interpolation. Processing time: 4.7 sec.



(a) (b) (c)

Figure 4. Implementation of the partition of unity for generation of polygons from scattered data of the fragment of Mount Bandai: (a) Curvature analysis. In the blue area, the surface variation $\sigma > 0.3$. (b) Result of reconstruction (ray tracing). Number of scattered points: 10000, processing time: 0.941 sec, number of vertices after reconstruction: 90000. (c) Fragment of the mesh as a wire-frame with color attributes according to calculated heights.



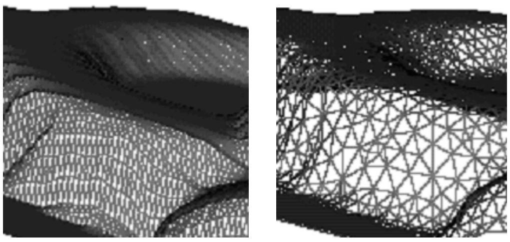
(a) (b) (c) (d)

Figure 5. Surface reconstruction of a technical data set. (a) Cloud of points (4100 scattered points are used). (b) Simplified mesh shaded (processing time: 0.1 sec). (c), (d) Fragments of the mesh as wire-frame (7141 vertices).

Figure 5 shows an implementation of the PUM for generation of polygonal surfaces for point sets represented by elevation data. We demonstrate the applicability of the approach to data homeomorphic to a disc; nevertheless, since a closed object can be partitioned into a collection of bordered patches homeomorphic to a disc, this is no serious restriction, as it was mentioned in Horman and Greiner [25].

It can be observed that the simplification algorithm demonstrates the smoothness inherited from the original data sets, and exhibits denser triangulation in the areas with higher curvatures (see Figure 5(c)). Actually, this is an advantage of our approach: instead of applying a complicated fairing technique we apply a rather simple local approximation. Nevertheless, despite the many practical applications of triangle meshes in CG, there are applications such as FEM where well-shaped triangulations are needed; in addition to the deterioration in the accuracy of calculations, speed may be sacrificed in some applications. Still, well shaped triangulations may in fact be useful, as was mentioned in [26]: “forty-odd years after the invention of the finite element method, our understanding of the relationship between mesh geometry, numerical accuracy, and stiffness matrix conditioning remains incomplete, even in the simplest cases.”

In spite of that the average aspect ratio for the mesh shown in Figure 5 is 1.46. Figure 5(d) demonstrates that there are many badly shaped triangles, especially in the almost vertical parts of the model.



(a) (b)

Figure 6. (a) Fragment of the initial mesh, 31234 triangles. (b) Combined mesh modification (polygon reduction and statistical improvement of the mesh), 12132 triangles.

There are many optimization techniques for optimal point placement, most of them based on the idea of local optimization. We are now investigating a statistical approach for triangles enhancement (a description of the algorithm is omitted because of constraints on space). Essentially, the algorithm can be used in combination with the edge collapse algorithm, which allows us to select new points more gently, as shown in Fig. 6 (b).

6. Conclusion

We have developed a set of algorithms to reconstruct and improve a surface obtained from a set of unorganized points. RBFs seem ready-made for many applications in CAD and CG, even for such applications as simplification and nearly interactive reconstruction of 3D models.

There is no single restoration and simplification method that provides the best results for every surface in the sense of quality and processing time. Experimental results indicate that the algorithms proposed in this paper provide rather good results and look promising for implementation in CAD and computer-aided engineering applications. It is proposed to use the bending energy $\boldsymbol{h}^t\boldsymbol{A}^{-1}\boldsymbol{h}$ as an error/quality cost to select candidates for edge collapse. More sophisticated optimization strategies, such as greedy optimization, can probably be combined with the proposed in the paper strategy. Let us note here that, for example, in Figure 6, the volume was rather well preserved, with a difference between the initial mesh and the processed one of about 0.65%. One shortcoming of the approach discussed in section 3 is that in some areas (almost vertical) of a surface the triangular vertices may be spaced far apart. Selecting local data sets in the reconstruction algorithm according to the surface curvature is still an interesting research topic waiting for a solution. Our most urgent problem is to extend the algorithm given in section 3 to provide adaptive remeshing (enriching) according to local features of a surface geometry. Another future plan is to provide a posterior error estimation by means of a finite element solution.

7. References

[1] P. J. Frey, “About Surface Remeshing,” *9th International Mesh Roundtable*, Sandia National Laboratories, 2000, pp. 123-136.

[2] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silvia, “Point Set Surfaces,” *Proceedings of IEEE Visualization 2001*, 2001, pp. 21-23.

[3] M. Pauly, M. Gross, and L. Kobbelt, “Efficient Simplification of Point-Sampled Surfaces,” *Proceedings of IEEE Visualization 2002,* MA, USA, pp. 163-170.

[4] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface Reconstruction from Unorganized Points," *Proceedings of SIGGRAPH '92*, 1992, pp.71-78.

[5] N. Amenta, S. Choi, and R. Kolluri, "The Powercrust," *Proceedings of the 6th ACM Symposium on Solid Modeling and Applications*, Michigan, 2001, pp. 249-260.

[6] R. Franke and G. Nielson, "Smooth Interpolation of Large Sets of Scattered Data," I*nternational Journal of Numerical Methods in Engineering*, 15, 1980, pp. 1691-1704.

[7] R. K. Beatson and W. A. Light, "Fast Evaluation of Radial Basis Functions: Methods for 2-D Polyharmonic Splines," *Tech. Rep. 119*, Mathematics Department, Univ. of Canterbury, Christchurch, New Zealand, 1994.

[8] D. Shepard, "A Two-Dimensional Interpolation Function for Irregularly Spaced Data," *Proceedings of the 23th Nat. Conf. of the ACM*, 1968, pp. 517-523.

[9] J. H. Ahlberg, E. N. Nilson, and J. L. Walsh, *The Theory of Splines and Their Applications*, Academic Press, New York, 1967.

[10] J. Dushon, "Splines Minimizing Rotation Invariants Semi-norms in Sobolev Spaces," in W. Schempp and K. Zeller (eds), *Constructive Theory of Functions of Several Variables,* Springer-Verlag, 1976, pp. 85-100.

[11] V. A. Vasilenko, *Spline-functions: Theory, Algorithms, Programs,* Novosibirsk, Nauka Publishers, 1983.

[12] F. L. Bookstein, "Morphometric Tools for Landmarks Data," Cambridge University Press, 1991.

[13] V. Savchenko, A. Pasko, O. Okunev and T. Kunii, "Function Representation of Solids Reconstructed from Scattered Surface Points and Contours," *Computer Graphics Forum*, 14(4), 1995, pp. 181-188.

[14] G. Turk and J. F. O`Brien, "Shape Transformation Using Variational Implicit Functions," *Proceedings of SIGGRAPH `99*, 1999, pp. 335-342.

[15] Y. Ohtake, A. Belyaev, and H-P. Seidel, "A Multi-scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions," *Proceedings of SMI '2003*, Soul, Korea, 2003 (accepted).

[16] M. Garland, "A Multiresolution Modeling: Survey & Future Opportunities", *EUROGRAPHICS`99*, State of The Art Reports, 1999.

[17] J. Wu and L. Kobbelt, "Fast Mesh Decimation by Multiple-Choice Techniques," *Proceedings of Vision, Modeling, Visualization 2002*, 2002, pp. 241-248.

[18] D.A. Field, "Laplacian Smoothing and Delaunay Triangulations," *Communications in Applied Numerical Methods*, vol. 4, 1998, pp. 709-712.

[19] G. Taubin, "A Signal Processing Approach to Fair Surface Design," *Proceedings of SIGGRAPH '95*, 29, 1995, pp. 351-358.

[20] H. Wendland, "Piecewise Polynomial, Positive Defined and Compactly Supported Radial Functions of Minimal Degree," *AICM*, 4, 1995, pp. 389-396.

[21] J. M. Melenk and I. Babuska, "The Partition of Unity Finite Element Method: Basic Theory and Applicationsd," *Research Report No. 96-01*, Seminar fur Angewandete Mathematik Eidgenossische Technische Hochschule CH-8092, Zurich, Switzerland, 1996.

[22] P. Krysl and T. Belytchko, "An Efficient Linear-precision Partition of Unity Basis for Unstructured Meshless Methods," *Communications in Numerical Methods in Engineering*, 16, 2000, pp. 239-255.

[23] V. Savchenko and A. Pasko, "Transformation of Functionally Defined Shapes by Extended Space Mappings," *The Visual Computer*, 14, 1998, pp. 257-270.

[24] The Visualization Toolkit Textbook and Open Source C++ Library, with Tcl, Python, and Java bindings. http://www.kitware.com/vtk.html, published by Kitware, 2001.

[25] K. Hormann and G. Greiner, "Quadrilateral Remeshing," *Proceedings of Vision, Modeling and Vizualization*, 2000, pp. 153-162.

[26] J.R. Shewchuk, "What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures," *Proceedings of the 11th International Meshing Roundtable,* Sandia National Laboratories, 2002, pp. 115-126.