NUMERICAL MATHEMATICS: Theory, Methods and Applications
Numer. Math. Theor. Meth. Appl., Vol. **2**, No. 1, pp. 90-99 (2009)

Adaptive Surface Reconstruction Based on Tensor Product Algebraic Splines

Xinghua Song and Falai Chen*

Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, China.

Received 13 December 2007; Accepted (in revised version) 26 September 2008

Abstract. Surface reconstruction from unorganized data points is a challenging problem in Computer Aided Design and Geometric Modeling. In this paper, we extend the mathematical model proposed by Jüttler and Felis (Adv. Comput. Math., 17 (2002), pp. 135-152) based on tensor product algebraic spline surfaces from fixed meshes to adaptive meshes. We start with a tensor product algebraic B-spline surface defined on an initial mesh to fit the given data based on an optimization approach. By measuring the fitting errors over each cell of the mesh, we recursively insert new knots in cells over which the errors are larger than some given threshold, and construct a new algebraic spline surface to better fit the given data locally. The algorithm terminates when the error over each cell is less than the threshold. We provide some examples to demonstrate our algorithm and compare it with Jüttler's method. Examples suggest that our method is effective and is able to produce reconstruction surfaces of high quality.

AMS subject classifications: 65D17

Key words: Surface reconstruction, algebraic spline surface, adaptive knot insertion.

1. Introduction

With the development of modern industry, it is possible to capture extremely large unorganized data points from the surfaces of existing models and products. Meanwhile, reproduction of existing models and products with complex free-form surfaces plays a very important role in CAD/CAM, Computer Vision, Computer Graphics, etc. The significance of surface reconstruction from point clouds attracts many researchers to investigate efficient and robust algorithms to solve the problem.

Surface reconstruction has been widely studied since the 1980s. A class of approaches in parametric surface reconstruction are based on the active contour models which were first proposed in [12] to detect image contours. Pottmann et al. [15] applied the technique to surface approximation, and they proposed an active parametric B-spline model

*Corresponding author. *Email addresses:* xhsong@mail.ustc.edu.cn (X. Song), chenfl@ustc.edu.cn (F. Chen)

to fit smooth given curves and surfaces [14]. Wang et al. [21] extended their work to the case of unorganized point cloud data and improved the efficiency of Pottmann's method dramatically. Recently, an evolution-based least square fitting method was also put forward to fit point clouds [2] and curves [18]. However, parametric fitting methods are difficult to handle point clouds with complicated topology. And also, parametric surface reconstruction needs a process of parametrization which is a non-trivial problem.

To solve the difficulty of parametric curve/surface reconstruction, implicit representation has been introduced. Carr et al. [4] introduced polyharmonic radical basis functions and multi-pole methods to model large data sets by a single radical basis function. Zhao et al. [26,27] applied the level set method in surface reconstruction by solving a PDE equation numerically. Their approach becomes very expensive both in time and in usage of memory when high accuracy reconstruction is required. Alexa et al. [1] developed the projection-based approaches, which have the advantage that they are local and they directly yield a point on the surface. Their approach requires the solution of a non-linear moving least square problem in a projection set, which makes many geometrical operations expensive.

The signed distance function has been used to reconstruct an implicit surface on a rectangular grid with the signs to distinguish inside and outside of the surface [3, 7, 8]. Ohtake et al. [13] proposed a hierarchical approach for 3D scattered data interpolation with compactly supported basis functions. Although this approach can process very large set of points, the implicit function used does not have an explicit form which is critical to theoretical analysis, such as multi-resolution analysis, approximation error, etc. Jüttler and Felis [10] described a technique for fitting surfaces to scattered data by simultaneously approximating points and associated normal vectors which are estimated from the given data. This approach is quite efficient due to a simple representation using algebraic B-spline functions with fixed meshes and a linear optimization method. However, the method can not get satisfactory result when the data points contain rich local geometric details. Other work includes moving least squares [6], dynamic implicit surface reconstruction [22, 24, 25], etc.

In this paper, we extend the mathematical model proposed by Jüttler and Felis in [10] from fixed tensor-product meshes to adaptive tensor-product meshes. The basic idea is as follows. We start with an initial mesh, over which the model proposed by Jüttler is applied to obtain an initial fit to the given point cloud. Then we check the fitting errors over each cell, and insert new knots in cells over which the errors are larger than some given threshold, and reconstruct a new algebraic surface to locally fit the given data. This process is recursively applied until the errors over each cell of the mesh are less than the given threshold. Our approach can produce reconstruction surfaces with much higher quality and is much more efficient than Jüttler's method due to adaptive meshes generation.

The rest of the paper is organized as follows. In Section 2, the algebraic B-spline surfaces are briefly introduced. In Section 3, we review the optimization reconstruction model proposed by Jüttler and Felis in [10] and extend it to adaptive meshes in the next section. In Section 5, we demonstrate our new algorithm with some examples and compare it with the technique presented in [10]. Finally, we conclude the paper by proposing some problems for future research in Section 6.

2. Algebraic tensor-product B-spline surfaces

Let $f(x,y,z)$ be a trivariate tensor product spline function of a tri-degree (m',n',l') defined over some domain Ω:

$$f(x,y,z)=\sum_{r,s,t} c_{rst}M_r(x)N_s(y)L_t(z), \tag{2.1}$$

where $\{M_r(x)\}_{r=1}^m, \{N_s(y)\}_{s=1}^n and \{L_t(z)\}_{t=1}^l$ denote the B-spline functions of degree m',n' and l' with some given knot sequences respectively, and c_{rst} denote the real control coefficients. The zero set of the function is defined by

$$V(f)=\{(x,y,z)\in\Omega | f(x,y,z)=0\}, \tag{2.2}$$

and it is called an *algebraic B-spline surface*.

For simplicity of notation, the control coefficients and the basis functions are gathered into two column vectors, denoted by

$$\mathbf{f}=(f_{111},\cdots,f_{mnl})^T \quad \text{and} \quad \mathbf{q}(\mathbf{P})=(\cdots,M_r(x)N_s(y)L_t(z),\cdots),$$

respectively, where $\mathbf{P}=(x,y,z)$. Then we can rewrite $f(x,y,z)$ (or $f(\mathbf{P})$) in the form

$$f(\mathbf{P})=\mathbf{q}(\mathbf{P})^T\mathbf{f}=\mathbf{q}^T\mathbf{f}, \tag{2.3}$$

and express the gradient $\nabla f(\mathbf{P})$ as

$$\nabla f(\mathbf{P})=\left(\frac{\partial f}{\partial x}(\mathbf{P}),\frac{\partial f}{\partial y}(\mathbf{P}),\frac{\partial f}{\partial z}(\mathbf{P})\right)^T=\begin{pmatrix}\mathbf{u}^T\mathbf{f}\\ \mathbf{v}^T\mathbf{f}\\ \mathbf{w}^T\mathbf{f}\end{pmatrix}, \tag{2.4}$$

where

$$\mathbf{u}=\frac{\partial\mathbf{q}}{\partial x},\quad \mathbf{v}=\frac{\partial\mathbf{q}}{\partial y},\quad \mathbf{w}=\frac{\partial\mathbf{q}}{\partial z}.$$

3. The optimization model with fixed tensor product meshes

Given data points $\{\mathbf{P_u}\}_{u=1}^U$ and the associated normal vectors $\{\mathbf{n_u}\}_{u=1}^U$ (If the normal vectors are not available, they can be approximately calculated from the given data set), we want to construct an algebraic spline surface to fit the data points. Jüttler presented an optimized model to handle this problem in [10]. The basic idea is as follows.

The given data points are approximated with an algebraic surface $f(x,y,z)=0$ by minimizing the sum of the squared algebraic distance

$$L(\mathbf{f})=\sum_{u=1}^{U}[f(\mathbf{P}_u)]^2=\sum_{u=1}^{U}\mathbf{f}^T\mathbf{A}_u\mathbf{f}, \tag{3.1}$$

where $\mathbf{f}$ is the coefficient vector of $f(x,y,z)$ and $\mathbf{A}_u = \mathbf{q}_u\mathbf{q}_u^T$.

Obviously, the minimum value of $L(\mathbf{f})$ is zero when $f(x,y,z) \equiv 0$, which produces meaningless result. To solve this problem, we add the normal deviation term by minimizing (in addition to algebraic distance term) the sum

$$\begin{aligned} N(\mathbf{f}) &= \sum_{u=1}^{U} \|\nabla f(\mathbf{P}_u) - \mathbf{n}_u\|^2 \\ &= \sum_{u=1}^{U} \left[\mathbf{f}^T\mathbf{B}_u\mathbf{f} - 2(n_{ux}\mathbf{u}^T\mathbf{f} + n_{uy}\mathbf{v}^T\mathbf{f} + n_{uz}\mathbf{w}^T\mathbf{f}) + \mathbf{n}_u^T\mathbf{n}_u\right], \end{aligned} \tag{3.2}$$

where

$$\mathbf{B}_u = \mathbf{u}_u\mathbf{u}_u^T + \mathbf{v}_u\mathbf{v}_u^T + \mathbf{w}_u\mathbf{w}_u^T, \quad \mathbf{n}_u = (n_{ux}, n_{uy}, n_{uz})^T.$$

The minimization of weighted linear combination of the algebraic distance and the normal deviation term should provide a solution to the fitting problem. However, such solution may contain extraneous surface components. To eliminate such situation, we further add a tension term $G(\mathbf{f})$ in the objective function:

$$\begin{aligned} G(\mathbf{f}) &= \iiint_{\Omega} \left(f_{xx}^2 + f_{yy}^2 + f_{zz}^2 + 2f_{xy}^2 + 2f_{yz}^2 + 2f_{zx}^2\right) dxdydz \\ &= \mathbf{f}^T\mathbf{H}\mathbf{f}, \end{aligned} \tag{3.3}$$

where $\mathbf{H}$ is a matrix of same size with $\mathbf{A}_u$ and $\mathbf{B}_u$. Finally, we will solve the following optimization problem:

$$F(\mathbf{f}) = L(\mathbf{f}) + \omega_1 N(\mathbf{f}) + \omega_2 G(\mathbf{f}) \to \text{Min.}, \tag{3.4}$$

where ω_1, ω_2 are non-negative weights.

The solution of the optimization problem (3.4) can be obtained by solving the following system of linear equations

$$\frac{\partial}{\partial \mathbf{f}} F(\mathbf{f}) = 0. \tag{3.5}$$

Moreover, if we substitute (3.1-3.4) into (3.5), we can write the linear equations (3.5) in a matrix form:

$$\mathbf{M}\mathbf{f} = \mathbf{b}, \tag{3.6}$$

where

$$\begin{aligned} \mathbf{M} &= \sum_{u=1}^{U} (\mathbf{A}_u + \omega_1\mathbf{B}_u) + \omega_2\mathbf{H}, \\ \mathbf{b} &= \omega_1 \sum_{u=1}^{U} (n_{ux}\mathbf{u}_u + n_{uy}\mathbf{v}_u + n_{uz}\mathbf{w}_u). \end{aligned}$$

4. The optimization model with adaptive tensor product meshes

For surface models with very complex topology and rich geometric details, it is difficult to construct high quality object surfaces in a reasonable amount of time by the mathematical model proposed in the last section, since generally a uniform mesh is adopted, and the size of the linear system of equations (3.6) is proportional to rst and hence the mesh can't be too dense. In order to solve this problem, we propose an adaptive mesh surface reconstruction technique based on the model in the above section. This method takes the rectangular bounding box of the data points as the initial mesh, and a reasonable tensor-product mesh is obtained automatically through an adaptive knot insertion process. Since in each step a local fitting problem is solved, the algorithm is much more efficient and high quality surface construction becomes possible.

To be concrete, we start from the rectangular bounding box (level '0' tensor product B-spline mesh T_0) of the data point set $\mathscr{P} = \{\mathbf{P}_u\}_{u=1}^{U}$, and construct an initial algebraic spline surface $f_0(x,y,z) = 0$ to fit the data points by the Juettler's method. Suppose we have constructed an algebraic spline surface $f_n(x,y,z) = 0$ over the level n tensor product B-spline mesh T_n. Then the level $n+1$ tensor product B-spline mesh T_{n+1} can be obtained according to the adaptive knot insertion algorithm which is based on the approximation errors in the cells of T_n and will be discussed in the next section. We then modify the spline function $f_n(x,y,z)$ by a displacement function $g_n(x,y,z)$ such that

$$f_{n+1}(x,y,z) := f_n(x,y,z) + g_n(x,y,z) = 0$$

is a better approximation to the point clouds than $f_n(x,y,z) = 0$. Specifically, let $\mathscr{D}$ be the union of all the cells in T_n, over which the approximation errors are larger than some threshold, and let $g_n(x,y,z)$ be a linear combination of B-spline basis functions whose supports have intersection with $\mathscr{D}$. Compute $g_n(x,y,z)$ by solving the optimization problem

$$\begin{aligned}\min F(g_n) = &\sum_{\mathbf{P}_u \in \mathscr{P} \bigcap \mathscr{D}} \Big[((f_n+g_n)(\mathbf{P}_u))^2 + \omega_1 \|\nabla(f_n+g_n)(\mathbf{P}_u) - \mathbf{n}_u\|^2\Big] \\ &+ \omega_2 \iiint_{\Omega} \Big[(f_{nxx}+g_{nxx})^2 + (f_{nyy}+g_{nyy})^2 + (f_{nzz}+g_{nzz})^2 \\ &+ 2(f_{nxy}+g_{nxy})^2 + 2(f_{nyz}+g_{nyz})^2 + 2(f_{nzx}+g_{nzx})^2\Big] dx\, dy\, dz. \end{aligned} \tag{4.1}$$

Denote the coefficient vector of function $g_n(x,y,z)$ by $\mathbf{g}$. Then $F(g_n)$ can be written in a quadratic form $F(\mathbf{g})$. The optimal problem (4.1) can be solved from the linear system of equations:

$$\frac{\partial}{\partial \mathbf{g}} F(\mathbf{g}) = 0. \tag{4.2}$$

Set

$$f_{n+1}(x,y,z) = f_n(x,y,z) + g_n(x,y,z).$$

In this way, we can recursively find the next level of algebraic spline surface which is a better approximation to the given point clouds.

5. The adaptive surface reconstruction algorithm

According to the optimization model in last section, we present the adaptive algorithm of surface reconstruction as follows.

Step 1 Input the data point set $\mathscr{P} = \{\mathbf{P}_u\}_{u=1}^U$ and the associated normal vectors $\{\mathbf{n_u}\}_{u=1}^U$. Find the bounding box of the point set as the initial tensor product B-spline mesh T_0. Solve the linear system (3.6) to get the initial algebraic spline surface $f_0(x,y,z) = 0$. Choose a threshold $\tau > 0$ for approximation errors and set $n = 0$.

Step 2 Compute the average approximating error

$$\overline{\varepsilon}_C = \frac{1}{\#(\mathscr{P} \bigcap C)} \sum_{\mathbf{P}_u \in \mathscr{P} \bigcap C} \frac{|f_n(\mathbf{P}_u)|}{\|\nabla f_n(\mathbf{P}_u)\|}$$

for each cell C of T_n which contains at least L points. Here L is selected to avoid subdividing cells with a few data points, since wherein the noise may play an important role. If $\overline{\varepsilon}_C \leqslant \tau$ for every cell C in T_n, then terminate the process; otherwise, let $\mathscr{D}$ denote the union of all the cells of T_n, over which the average of approximating errors are larger than the given threshold τ. Perform knot insertions along x, y, z directions such that each cell in $\mathscr{D}$ is subdivided into eight subcells with the same size. The mesh after knot insertion is denoted by T_{n+1}. Compute the coefficient vector $\mathbf{f}$ of the tensor product B-spline function $f_n(x,y,z)$ over the new mesh T_{n+1} using Bohm's algorithm.

Step 3 Let $g_n(x,y,z)$ be a linear combination of all the basis functions whose supports intersect with $\mathscr{D}$. Denote the support of $g_n(x,y,z)$ by $\mathrm{SP}(g_n)$. Solve the optimization problem (4.1) and obtain a displacement function $g_n(x,y,z)$.

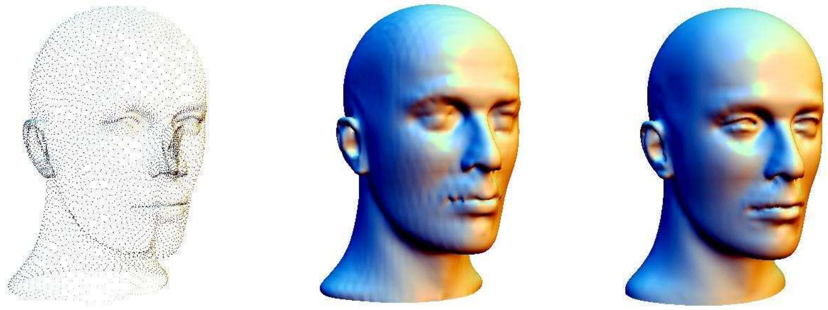
Step 4 Set $f_{n+1}(x,y,z) = f_n(x,y,z) + g_n(x,y,z)$, and let $n = n+1$. Return to Step 2.

In each step, the optimization problem (4.1) is solved over the region $\mathscr{D}$ where the approximation error is larger than the threshold τ, thus the size of the linear system (4.2) decreases dramatically.

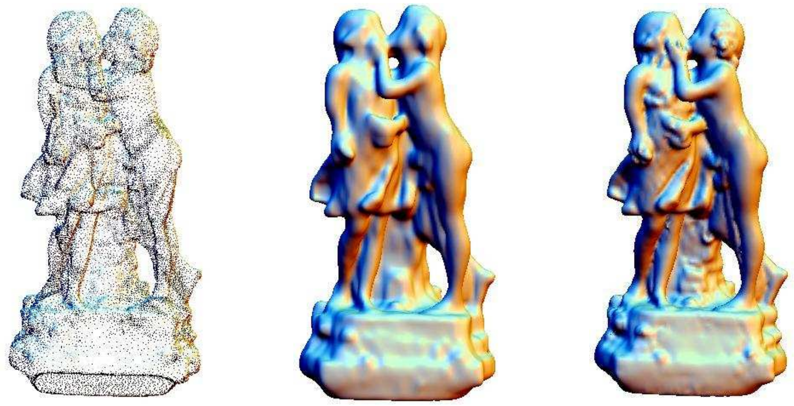
6. Implementation and examples

In this section, we illustrate several examples to demonstrate our algorithm. Comparison is also made between our method and Jüttler's method.

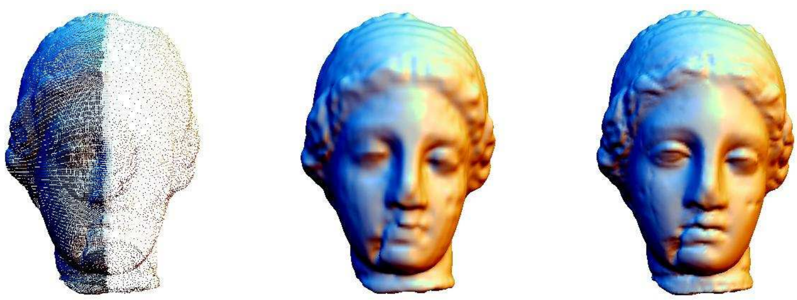
The algorithm presented in [10] needs the user to set a fixed prior partition of the tensor product mesh. However, how to set the number of knots in each direction totally depends on the user's experience, and different choice of mesh partition may influence the



(a) data points (b) The method of [10] (c) Our method



(a) data points (b) The method of [10] (c) Our method



(a) data points (b) The method of [10] (c) Our method

Figure 1: Comparison of the results obtained by using the method of [10] and the method proposed in this work: Mannequin (top), Sculpture (middle), and HalfVenus (bottom).

quality of the reconstruction surface. On the other hand, by our adaptive knot insertion algorithm, the user does not need to specify the partition of the tensor product mesh. A reasonable tensor product mesh can be obtained automatically during the iterative process.

As we mentioned before, it is difficult to fit the data points with local geometric details very well by solving the optimization reconstruction model based on a fixed tensor-product

Table 1: Comparison between Juettler's method and our method.

Examples	Points	Partitions1	Time1	Partitions2	Time2
Mannequin	6737	$51\times44\times40$	11.6s	$102\times70\times60$	15.2s
Sculpture	29171	$83\times67\times72$	111.5s	$143\times125\times143$	91.9s
Venus	134345	$85\times97\times92$	140.4s	$101\times129\times129$	123.8s

mesh. Sparse knot sequences can not exhibit the local features, while dense knot sequences are superfluous and result in low efficiency. In the worst case, the algorithm proposed in [10] may terminate with unexpected result since the matrices in the model are too large to be saved in computer memory. On the other hand, with our local refinement strategy, only a much smaller matrix need to construct during the optimization process, and much fewer time is needed to construct the surface of same quality.

We implement several examples to compare our algorithm with the one proposed in [10].

The method of [10] restricts the sum in (2.1) to a certain subset of the index set. That subset corresponds to the boxes defined by the knot vectors which contain data, and to the neighbouring cells. Usually this strategy can make their algorithm more efficient and we adopt this strategy in the implementation.

Table 1 compares the computational statistics of the method in [10] and our method. The table items of 'Partitions1' and 'Times1' represent the statistics of the method of [10] while the 'Partitions2' and 'Times2' represent ours. All examples are implemented on a PIV-1.73GHz PC with 1.0GB RAM. The results are illustrated in Fig. 1. From the statistics and the figures, we conclude that our method is not only more efficient but also produces reconstruction surfaces with much higher quality than that of [10].

7. Conclusions

In this paper, we extended the surface reconstruction model proposed in [10] to adaptive meshes. An iterative algorithm is proposed to fit a given data sets with high quality reconstruction surfaces effectively. Several examples are provided to demonstrate the effectiveness of our algorithm. Compared with the method presented in [10], our technique can produce reconstruction surfaces with higher quality and can also generate a reasonable tensor product mesh automatically.

Due to the limit of tensor product structure of algebraic spline surfaces, there are superfluous control coefficients in the expressions of the algebraic spline surfaces. One possible improvement is to replace tensor product splines with T-splines [16] or splines over T-meshes [5]. The works of [24,25] have produced such an approach along this direction.

Acknowledgments This work is supported by the National Key Basic Research Project of China (No. 2004CB318000), One Hundred Talent Project of the Chinese Academy of Sciences, the NSF of China (No. 60225002, No. 60533060), Doctorial Program of MOE of

China and the 111 Project (No. B07033). The authors are grateful for the comments and suggestions by the anonymous reviewers.

References

[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, *Point set surfaces*, Proceedings of the IEEE Visualization Conference, pp. 21–28, 2001.

[2] M. Aigner, Z. Sir, and B. Jüttler, *Evolution-based least-squares fitting using Pythagorean hodograph spline curves*, Computer Aided Geometric Design, 24 (2007), pp. 310–322.

[3] C.L. Bajaj, F. Bernardini, and G. Xu, *Automatic reconstruction of surfaces and scalar fields from 3D scans*, SIGGRAPH'95 Conference Proceedings, pp. 109–118, July 1995.

[4] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, and T.R. Evans, *Reconstruction and representation of 3D objects with radial basis functions*, SIGGRAPH'01 Conference Proceedings, pp. 67–76, July 2001.

[5] J.S. Deng, F.L. Chen, X. Li, C.Q. Hu, W.H. Tong, Z.W. Yang and Y.Y. Feng, *Polynomial splines over hierarchical T-meshes*, accepted by Graphical Models, 2008.

[6] S. Fleishman, D. Cohen-Or, and C.T. Silva, *Robust moving least-squares fitting with sharp features*, ACM Transactions on Graphics, 3 (2005), pp. 544–552.

[7] A. Hilton, J. Stoddart, A.J. Illingworth, and T. Windeatt, *Implicit surface-based geometric fusion*, Computer Vision and Image Understanding, 69 (1998), pp. 273–291.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, *Surface reconstruction from unorganized points*, SIGGRAPH'92 Conference Proceedings, pp. 71–78, July 1992.

[9] B. Jüttler, *Least-squares fitting of algebraic spline curves via normal vector estimation*, R.Cipolla, R.R.Martin, eds., The Mathematics of Surfaces IX, Springer-Verlag, pp. 263–280, 2000.

[10] B. Jüttler and A. Felis, *Least-squares fitting of algebraic spline surfaces*, Advances in Computational Mathematics, 17 (2002), pp. 135–152.

[11] E. Wurm and B. Jüttler, *Approximate implicitization via curve fitting*, Eurographics Symposium on Geometry Processing, pp. 240–278, 2003.

[12] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: active contour models*, International Journal of Computer Vision, pp. 321–331, 1988.

[13] Y. Ohtake, A. Belyaev, and H. P. Seidel, *Multi-scale approach to 3D scattered data interpolation with compactly supported basis functions*, Proceedings of Shape Modeling International, pp. 153–161, 2003.

[14] H. Pottmann, S. Leopoldseder, M. Hofer, *Approximation with active B-spline curves and surfaces*, Proceedings of Pacific Graphics, pp. 223–244, 2002.

[15] H. Pottmann and M. Hofer, *Geometry of the squared distance function to curves and surfaces*, H. Hege and K. Polthier, eds., Visualization and Mathematics III, pp. 223–244, 2003.

[16] T.W. Sederberg, J. Zheng, A. Bakenov and A. Nasri, *T-splines and T-NURCCs*, ACM Transactions on Graphics, 22(2003), pp.477–484.

[17] D. Shepard, *A two-dimensional interpolation function for irregularly-spaced data*, Proceeding 23rd ACM National Conference, pp. 517–524, 1968.

[18] Z. Sir, R. Feichtinger and B. Jüttler, *Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics*, Computer Aided Design, 38 (2006), pp. 608–618.

[19] W.H. Tong, F.L. Chen and Y.Y. Feng, *A fast and adaptive surface reconstruction algorithm based on the implicit tensor-product B-spline (ITPBS) surfaces*, in Proceedings of The Seventh China-Japan Seminar on Numerical Mathematics, pages 161–178, 2006.

[20] G. TRUK AND J.F. O'BRIEN, *Shape transformation using variational implicit functions*, Siggraph'99 Conference Proceedings, pp. 335–342, August 1999.

[21] W. WANG, H. POTTMAN, AND Y. LIU, *Fitting B-spline curves to point clouds by squared distance minimization*, ACM Transactions on Graphics, 25 (2006), pp. 214–238.

[22] Z.W. YANG, J.S. DENG, AND F.L. CHEN, *Fitting unorganized point clouds with active implicit B-spline curves*, The Visual Computer, 21 (2005), pp. 831–839.

[23] Z.W. YANG, X.H. SONG, J.S. DENG, F.L. CHEN, *Dynamic implicit surface reconstruction with adaptive knots insertion*, in preparation.

[24] H.P. YANG, B. JÜTTLER, O. SCHERZER, AND M. FUCHS, *Evolution of T-spline level sets with distance field constraints for geometry reconstruction and image segmentation*, Shape Modeling and Applications 2006, IEEE Press, pp. 247–252, 2006

[25] H.P. YANG AND B. J ÜTTLER, *Evolution of T-spline level sets for meshing non-uniformly sampled and incomplete data*, The Visual Computer, in press.

[26] H. ZHAO, S. OSHER, B. MERRIMAN, AND M, KANG, *Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method*, Computer Vision and Image Understanding, 80 (2000), pp. 295–314.

[27] H. ZHAO AND S. OSHER, *Visualization, Analysis and Shape reconstruction of Unorganized Data Sets*, S.Osher, N.Paragios eds., Geometric Level Set Methods in Imaging, Vision and Graphics, Springer-Verlag, 2002.