Patera,J., Skala,V.: Radial Basis Function method for iso-line extraction, EC&I 2004, Košice, VIENALA Press, 2004

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS
OF
THE TECHNICAL UNIVERSITY OF KOŠICE

PROCEEDINGS
of
THE SIXTH INTERNATIONAL SCIENTIFIC CONFERENCE

ELECTRONIC COMPUTERS and INFORMATICS ECI 2004

The conference is organized by
Department of Computers and Informatics

In co-operation with:
Slovak Society for Applied Cybernetics and Informatics
Czech and Slovak Society for Simulation

Sponsoring by:
SIEMENS Program & System Engineering s.r.o. Bratislava
Ing. Milan Roško, TEGH, Information Technology Dep. Toronto, Kanada
ZTS Výskumno-vývojový ústav a.s., Košice

Editors: Štefan Hudák, Ján Kollár

September 22-24, 2004
Košice - Herľany
Slovakia

Editors' Note

This publication was reproduced by the photo process, using the manuscripts and soft copies supplied by their authors. The layout, figures, and tables of some papers did not comform exactly with the standard requirements. In some cases the layout of the manuscripts were rebuilt. All mistakes in manuscripts there either could not be fixed, or could not be checked completely by reviewers and there are a responsibility of authors. The readers are therefore asked to excuse any deficiencies in this publication which may have arisen from the above causes.

Proceedings of the Sixth International Scientific Conference Electronic Computers and Informatics ECI 2004
ISBN 80-8073-150-0
Editors: Štefan Hudák, Ján Kollár
September 22-24, 2004, Košice - Herľany, Slovakia

© Layout & Design: J. Bača, D. Mihalyi, D. Sobotová

Additional copies can be obtained from:
Department of Computers and Informatics of FEI,
The University of Technology Košice,
Letná 9, 04200 Košice, Slovakia
Phone: +421-55-63 353 13
Fax: +421-55-6330115
E-mail: Stefan.Hudak@tuke.sk, Jan.Kollar@tuke.sk

Proceedings of
the Sixth International Scientific Conference
Electronics Computer and Informatics ECI 2004

VIENALA Press
Moldavská 8/A, 040 01 Košice
Edition: 55
September 2004

ISBN 80-8073-150-0

Uhlir,K., Patera,J., Skala,V.: Radial Basis Function method for iso-line extraction, EC&I 2004, Košice, VIENALA Press, 2004

Radial Basis Function method for iso-line extraction

Karel Uhlíř*, Jan Patera*, Václav Skala
Department of Computer Science and Engineering, Faculty of Applied Sciences,
University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic
E-mail: {kuhlir | hopatera | skala}@kiv.zcu.cz

Abstract

This paper presents analysis and comparison of the basic method for iso-line approximation with Radial Basis Function (RBF) method on 2D scalar data. Different aspects are taken into account and compared methods are described. The RBF is complex and computational expensive but can deal with scattered points. Any iso-line defined by these points can be represented by a functional description. Standard method for linear interpolation is simple but does not provide fine results for under sampled data. Also the input data must be ordered.

Keywords: RBF, implicit surface, iso-line, extraction, implicitization, comparison.

1. Introduction

In the last few years the methods for iso-surface or iso-line extraction are often used in many branches, e.g. in medicine for visualization of MRI (Magnetic Resonance Imaging) or CT (Computed Tomography) data, geography or in meteorology for visualization of weather simulation. The basic methods for iso-line (2D) or iso-surface (3D) extraction are well known such as linear, bicubic, bilinear or trilinear (3D) interpolations. They are used in methods for data visualization, e.g. marching cubes, marching tetrahedra or marching squares (2D) very often.

Modeling with implicitly defined objects is the part of computer graphics and computational geometry, which is rapidly developing now. It is a perfect description of objects. The object is defined by an implicit equation $f(x)=0$ and it can be directly visualized from its implicit form. The visualization of implicit surfaces typically consists of finding the zero-set of f, which may be performed either by polygonizing the surface [1], [4], [6] or by direct ray tracing [5]. There are a lot of techniques for visualization and rendering of the implicit surfaces that are defined by an implicit function $f(\mathbf{x})=0$ [2].

The problem is that there are not enough real objects that are defined by an implicit function. There are a few basic primitives, e.g. sphere, cylinder, torus that are used for modeling in CSG (Constructive Solid Geometry) together with Boolean operations. Penetration in this field is in a new method for implicitization of real objects based on variational implicit functions using the Radial Basis Function method (RBF). The RBF is a complex and computationally expensive method for interpolation of scattered data. See Section 2 for details.

Our idea is to use the RBF method for approximation of iso-line in CT data instead of standard methods and to find the implicit representation of the iso-line in the uniform grid. Results of our work are presented in Section 6. We concentrated on 2D for simplicity. We compared our solution with a standard method for iso-line extraction. This method is simple and fast and it is known as Marching Squares (MS). In this paper, we aimed at the use of RBF method without on-curve points to find zero iso-line and we compared this strong approach with standard iso-line algorithm.

1.1. Problem definition

The iso-line interpolation with the RBF method refers to the *scattered data interpolation* problem. The problem of scattered data interpolation is to create a smooth function that passes through a given set of data points [7].

* This work was supported by the Grant No.: MSM 235200005

The iso-line extraction problem can be defined as:

Problem: Given a surface S by n distinct points $x_1, \ldots, x_n$, where $x_i=[x_i, y_i, h_i]^T$, where value of h_i is defined as $h_i=f(x_i, y_i)$. The surface is approximated/interpolated by a function $z=F(x, y)$. For the given value q find an iso-curve L, for which $F(x, y)=q$.

In order to be able to asses RBF behavior we will compare RBF method with the linear interpolation method, see Section 6.

2. Radial Basis Function interpolation

Having defined the problem to be solved, let us now describe the RBF method. Scattered data interpolation can be achieved using radial basis functions centered at the set of n input points (constraints). Radial basis functions are circularly symmetric functions centered at a particular point. The RBF method may be used to interpolate a function with n points by using n radial basis functions centered at these points. The resulting interpolated function thus becomes:

$$f(\mathbf{x})=\sum_{j=1}^{n}\lambda_j\phi(\|\mathbf{x}-\mathbf{c}_j\|) \qquad (1)$$

where $\mathbf{c}_j$ are given locations of a set of n input points (constraints), λ_j are the unknown weights, $\mathbf{x}$ is a particular point and $\phi(\|\mathbf{x}-\mathbf{c}_j\|)$ is a basis function where $\phi(0)=0$ and $\|\mathbf{x}-\mathbf{c}_j\|=r_j$. It can be evaluated for the given radial r_j, defined by the difference of the point in which we want to evaluate this function and the constraint. There are some popular choices for the basis function, e.g. the thin-plate spline $\phi(r)=r^2\log(r)$, the Gaussian $\phi(r)=\exp(-cr^2)$, the multiquadric $\phi(r)=\sqrt{(r^2+c^2)}$, biharmonic $\phi(r)=|r|$ and triharmonic $\phi(r)=|r|^3$ splines.

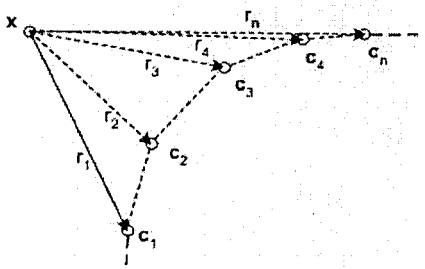


Fig. 1 – Solution of Eq. (1) for an arbitrary point $\mathbf{x}$

In order to find a set of λ_j for which the interpolation constraints satisfy $h_i=f(\mathbf{c}_i)$, we can define the right side of the Eq. (1) for $f(\mathbf{c}_i)$, which gives:

$$f(\mathbf{c}_i)=\sum_{j=1}^{n}\lambda_j\phi(\|\mathbf{c}_i-\mathbf{c}_j\|)=h_i \qquad (2)$$

For the given constraints $\mathbf{c}_i$ we can rewrite Eq. (1) to the form $h_i=f(\mathbf{c}_i)$ and Fig. 1 shows how the scalar values r_j for an arbitrary point $\mathbf{x}$ are defined.

Since this equation is linear with respect to the unknown λ_j it can be formulated as a system of linear equations. Let us define $\mathbf{c}_i=[c_i^x, c_i^y]^T$ and $\phi_{ij}=\phi(\|\mathbf{c}_i-\mathbf{c}_j\|)$ for an interpolation in 2D. Then this linear system can be written as follows:

$$\begin{bmatrix}\phi_{11} & \phi_{12} & \cdots & \phi_{1n}\\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn}\end{bmatrix}\begin{bmatrix}\lambda_1\\ \lambda_2\\ \vdots\\ \lambda_n\end{bmatrix}=\begin{bmatrix}h_1\\ h_2\\ \vdots\\ h_n\end{bmatrix} \qquad (3)$$

In some cases (including the thin-plate spline solution), it is necessary to add the first-degree polynomial $P(\mathbf{x})=ac^x+bc^y+c$ to account for linear and constant portion of f and to ensure positive definite of the system solution [3]. Then Eq. (1) is modified to the Eq. (4).

$$f(\mathbf{x})=\sum_{j=1}^{n}\lambda_j\phi(\|\mathbf{x}-\mathbf{c}_j\|)+P(\mathbf{x}) \qquad (4)$$

If a polynomial is included, Eq. (4) similarly becomes [8]:

$$\begin{bmatrix}\phi_{11} & \phi_{12} & \cdots & \phi_{1n} & c_1^x & c_1^y & 1\\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} & c_2^x & c_2^y & 1\\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots\\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} & c_n^x & c_n^y & 1\\ c_1^x & c_2^x & \cdots & c_n^x & 0 & 0 & 0\\ c_1^y & c_2^y & \cdots & c_n^y & 0 & 0 & 0\\ 1 & 1 & \cdots & 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\lambda_1\\ \lambda_2\\ \vdots\\ \lambda_n\\ a\\ b\\ c\end{bmatrix}=\begin{bmatrix}h_1\\ h_2\\ \vdots\\ h_n\\ 0\\ 0\\ 0\end{bmatrix} \qquad (5)$$

If we denote:

$$\begin{aligned}&\mathbf{A}_{i,j}=\phi(\|\mathbf{c}_i-\mathbf{c}_j\|), \quad i,j=1,\ldots,n\\ &\mathbf{p}_i=[c_i^x, c_i^y, 1]^T, \qquad i=1,\ldots,n\\ &\mathbf{a}=[a,b,c]^T\\ &\boldsymbol{\lambda}=[\lambda_1,\lambda_2,\ldots,\lambda_n]^T, \quad h=[h_1,h_2,\ldots,h_n]^T,\\ &\mathbf{P}=[\mathbf{p}_1\,|\,\mathbf{p}_2\,|\ldots|\,\mathbf{p}_n],\end{aligned} \qquad (6)$$

then we can rewrite the system in Eq. (5) to the form:

$$\mathbf{B}\begin{bmatrix}\boldsymbol{\lambda}\\ \mathbf{a}\end{bmatrix}=\begin{bmatrix}\mathbf{h}\\ \mathbf{0}\end{bmatrix},\ \text{where}\ \mathbf{B}=\begin{bmatrix}\mathbf{A} & \mathbf{P}^T\\ \mathbf{P} & \mathbf{0}\end{bmatrix} \qquad (7)$$

Afterwards, the linear equation system Eq. (5) is solved. Then the function $f(\mathbf{x})$ can be evaluated for an arbitrary point $\mathbf{x}$.

Scattered interpolation problem accounts from the fact that in the basic set of input points there are points on the curve and outside of the curve. On-curve points have zero value and off-curve points have non-zero value. In our case the basic set contains only off-curve points and we try to find the zero curve (iso-line).

Discussion on the system (7) solution and the variables determination will be provided in the following section.

3. System solution

If a basis function ϕ is known then we know the values of all elements of the matrix $\mathbf{B}(m \times m)$, where $m=n+3$ (n + rank of $\mathbf{a}$). It is obvious that condition $h_i=0$ is satisfied for all on-curve points thus $\mathbf{h}$ is a zero vector. Because we have no on-curve points thus all values in $\mathbf{h}$ vector are non-zero. The system matrix $\mathbf{B}$ has at the main diagonal zero elements and non-diagonal elements are non-zero. It can be verified that the rank of our system matrix $\mathbf{B}$ is equal to m and its determinant is non-zero. Thus the linear system can be solved.

It should be noted that LU factorization [7], [9], [8] is the most commonly used method for solving the linear system defined by the Eq. (7).

4. Input data

The data in 2D grid can represent, e.g. elevation, temperature, density, etc. In our case, the value of data samples represents their distance from zero contour $f(x,y)=0$. The value of samples is generated from the mathematic description – Eq. (8). Such an evaluated grid serves as an input for both mentioned methods (RBF and MS).

We used an ellipse for grid values generation in our tests.

$$f(x,y)=\frac{(x-s_x)^2}{a^2}+\frac{(y-s_y)^2}{b^2}-1 \qquad (8)$$

where x, y are grid vertex coordinates; s_x, s_y are a centre coordinates; a and b are ellipse axes.

Grid vertices that lie outside of the ellipse have positive, on the ellipse zero and inside negative ones. The values represent the distance between a grid vertex and zero contour of an ellipse.

5. Zero level tracking

5.1. RBF method

The RBF steps for zero level tracking are as follows:

1. At first, we select neighbors we want to use (4, 8, 12 and 16, see Fig. 2) and all cells are processed sequentially.
2. We have a set of points for each cell from a grid (depending on previously selected neighbors), points do not need to be ordered.
3. For the set of points, the RBF is evaluated Eq. (5).
4. Zero iso-line is approximated within a cell with the use of a step method (see Fig. 3).

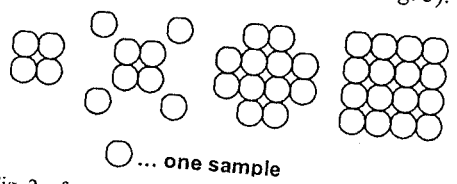


Fig. 2 – from the left 4, 8, 12 and 16 neighbor samples in a 2D grid

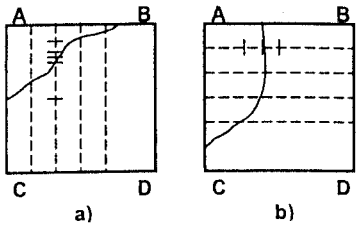


Fig. 3 – Step method examples: a) vertical steps, b) horizontal steps

For four neighbors the RBF generates almost the same iso-line as a linear approximation, see Figure 6a. For all λ stands $\lambda \approx 0$. Coefficients a, b, c of the polynomial $P(\mathbf{x})$ are non-zero. Hence, the impact of $\phi(r)$ is minimal.

5.2. Marching squares

Linear interpolation is typically used for an iso-line extraction from a 2D scalar data and it is known as MS method. It is based on the same principle as Marching Cubes [7]. The input of MS method has to be ordered to the regular grid. The output is a set of line segments that approximate the iso-line for the given threshold value. We are interested in the zero threshold.

The steps of MS method are as follows:

1. Sequential traversal of all the cells
2. The four bit index computation for the actual cell. The value of A, B, C, D nodes can be positive 1 or negative 0 => binary index = $ABCD_B$, see Fig. 24.
3. On the basis of the index we can decide which edges of the actual cell are intersected by the zero iso-line (with the use of a *special table*).
4. The linear interpolation is than used to find the intersections of the iso-line with the edges of the currently processed cell:

$$x_P = x_A + \frac{x_B - x_A}{B - A}(Threshold - A) \quad (9)$$

where x_P is the intersection point, x_A, x_B are edge end points, and A, B are data values in the x_A, x_B points.

The algorithm ends when all cells were traversed and the iso-lines can be drawn.

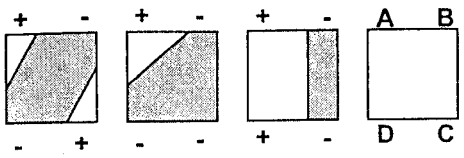


Fig. 4 – 3 basic Marching Squares cases and the cell description

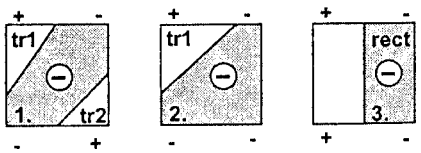
A *special table* created in the preprocessing contains all possible variants (16 cases) of how the iso-line can intersect cell edges. For each index configuration, the table also contains intersected edges. All 16 possibilities can be reduced to three basic cases due to rotation, see Fig. 4.

6. Methods comparison

The MS method (linear interpolation) and the RBF method are compared in this section. The comparison is done as to iso-line length and area, which is enclosed by the iso-line. In addition, the reasons and observations are discussed as well.

The iso-line length is computed as a sum of lengths of all linear segments of the iso-line (for both methods).

The area enclosed by this iso-line is computed as an area of a general polygon. Such an area cannot be computed directly, but we can take a cell area and subtract a triangle from it (depending on the computed bit index) to simplify the computations, see Fig. 5.



1. S=S(cell) - S(tr1) - S(tr2)
2. S=S(cell) - S(tr)
3. S=S(rect)

Fig. 5 – An area computation

The step method interpolates the zero level with line segments and thus the area computation is similar. The only difference is that the step method creates a set of trapezoids instead of one triangle that is produced by the MS method.

The area is summed up for all cells from the input data set. Of course that the cells which are not intersected by an iso-line have a zero contribution and the cells which are all inside the polygon have a maximum contribution to the final area.

The computed length and area are compared to the original function that was used to generate 2D grid data, in our case the ellipse.

$$S = \pi ab; \qquad l = b * \mathrm{E}(t,k); k = \sqrt{1 - \frac{a^2}{b^2}} \quad (10)$$

where S is the ellipse area; l is the ellipse length; $\mathrm{E}(t,k)$ is an incomplete elliptic integral of the second kind; a, b are ellipse axes (major and minor).

The radial basis function $\phi(r) = |r|^3$ and the ellipse with $a=2$, $b=1$ were used for all below presented tables and figures.

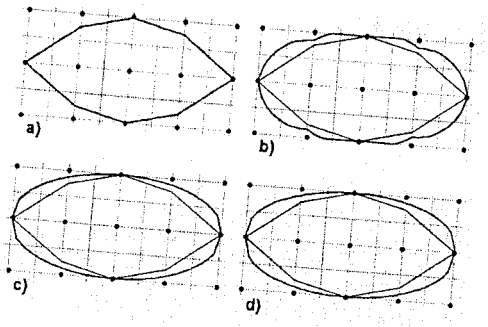


Fig. 6 – The RBF and linearly approximated contours for a) 4, b) 8, c) 12 and d) 16 neighbors in 2D. Dots show data samples.

Fig. 6 shows the impact of the number of neighbors to the final contour approximation. The RBF contour waving in Fig. 6b is due to 8 neighbor distribution (Fig. 2). Because the impact of cross points in corners is significant, then the sum of line segments is higher then length of original function (Tab. 1).

The 3D view of RBF patches and a zero contour extracted by MS method are shown in Fig. 7. In this case, for 4 neighbors the resulting image for the MS method is the same.

Note that all patches are almost planar for four neighbors. In Fig. 8 we want to show solution for the MS method (iso-line) and 16 neighbors (patches) case.

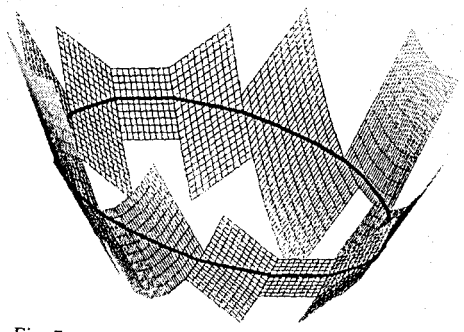


Fig. 7 – Linear interpolation and RBF patches (4 neighbors) in a 3D view

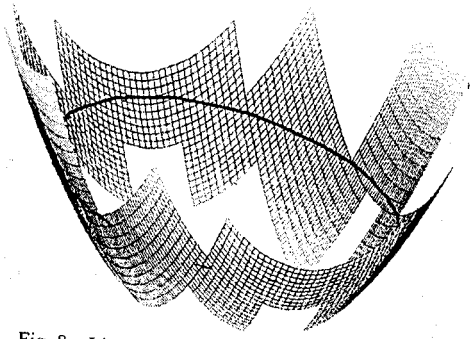


Fig. 8 – Linear interpolation and RBF patches (16 neighbors) in a 3D view

Both methods (RBF and MS) are compared as to the iso-line length and an area. In Tab. 1 the length of iso-lines are compared. Implicit equation $f(\mathbf{x}) = 0$ represents the exact analytical solution. The number of neighbors is important only for the RBF method and has no use in connection with MS method.

	4	8	12	16
exact	9.688	9.688	9.688	9.688
MS	9.123	9.123	9.123	9.123
RBF	9.138	9.873	9.811	9.770

Tab. 1 – Iso-curve length (for different number of neighbors)

For four neighbors the RBF method provides similar results as MS method. For 8, 12 and 16 neighbors the approximated contour length converges from above to the MS method output as well as to the original implicit function $f(\mathbf{x})=0$. The same holds for the area approximation as can be observed from Tab. 2.

	4	8	12	16
exact	6.28	6.28	6.28	6.28
MS	5	5	5	5
RBF	5.00	6.48	6.41	6.35

Tab. 2 – Enclosed area of iso-curve (for different number of neighbors)

	Method				
Resolution	MS	RBF 4	RBF 8	RBF 12	RBF 16
10x10	0.2	2.4	2.6	3.1	4.2
50x50	2.4	8.8	12.4	16.4	23.3
100x100	9.7	19.8	25.8	39.7	45.0

Tab. 3 – Iso-curve extraction times [s] (for different grid resolutions)

Comparison of times for different space divisions can be seen in Tab. 3. It is necessary to note, that time of computation is not quite relevant as computation was made in Matlab.

7. Conclusions

When comparing RBF method with Marching Squares method, the MS method can be substituted with RBF function, which provides similar results for four neighbors. With neighbor extension the better accuracy is reached. In addition, better continuity than C^0 (linear interpolation) is reached with the use of RBF method. The RBF method does not need ordered input points.

8. Future work

At first, we would like to improve and optimize the method we used for the RBF zero level tracking. The RBF interpolation could be also useful in image processing field to shrink or extrapolate images.

Also the extension of presented approach to 3D is another goal. Such approach could be used to an iso-surface extraction.

Acknowledgements

We are grateful to Vítězslav Adámek for his helpful information concerning mathematics and Matlab and also to Petra Jurisová for her patience.

References

[1] Bloomenthal, J.: Polygonizaion of Implicit Surface, Computer-Aided Geometric Design, vol. 5, no. 4, pp. 341-355, 1988

[2] Bloomenthal, J., Bajaj, C., Blinn, J., Cani-Gascuel, M. P., Rockwood, A., Wyvill, B., Wyvill, G.: Introduction to Implicit Surfaces, Morgan Kaufmann, 1997

[3] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., Evans, T. R.: Reconstruction and Representation of 3D Objects with Radial Basis Functions, Computer Graphics (SIGGRAPH 2001 proceedings), pp. 67-76, August 2001

[4] Čermák M.,Skala V.:Polygonization by the Edge Spinning, Algoritmy 2002 Conference proceedings, University of Technology, Slovakia, ISBN 80-227-1750-9, pp.245-252, 2002

[5] Hart, J.C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces, The Visual Computer 12 (10), Dec. 1996, pp. 527-545

[6] Hartmann, E.: A marching method for the triangulation of surfaces, The Visual Computer 14, 3, 95-108, 1998

[7] Lorensen,W.E., Cline, H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm, Computer Graphics, Volume 21, Number 4, July 1987

[8] Morse, B., Yoo, T. S., Rheingans, P., Chen, D. T., Subramanian, K. R.: Interpolating Implicit Surfaces from Scattered Surface Data Using Compactly Supported Radial Basis Functions, in Proceedings of the Shape Modeling conference, Genova, Italy, 89-98, May 2001

[9] Turk, G., O'Brien, J.F.: Modelling with Implicit Surfaces that Interpolate, ACM Transactions on Graphics, Vol. 21, No. 4, pp. 855-873, October 2002