



Crack-Free Isosurface of Volumetric Scattered Data

Han Sol Shin¹, Jee Ho Song¹, Tae Jun Yu², and Kun Lee³(✉)

¹ Department of Information and Communication,
Handong Global University, Pohang, Republic of Korea

² Department of Advanced Green Energy and Environment,
Handong Global University, Pohang, Republic of Korea

³ School of Computer Science and Electronic Engineering,
Handong Global University, Pohang, Republic of Korea
kunlee@handong.edu

Abstract. Isosurface extraction is the method of visualizing multivariate data in three-dimensional space. It is an important process to observe geometrical distribution of iso-value by isosurface extraction. Data obtained by PIC (Particle In Cell) simulation have a characteristic of irregularly distributed volumetric scattered data. Unlike curved surfaces of implicit function, PIC simulation data cannot be represented by continuous function, and each points have no relationship. In such case, isosurface extraction algorithms do not guarantee crack-free surfaces on their results. This paper describes how we get smooth approximation of volumetric scattered data by using a natural neighbor interpolation, and extract a crack-free isosurface on interpolated data.

Keywords: Isosurface · Volumetric scattered data · Data visualization

1 Introduction

Visualization of three- or more dimensional data is a major problem of data visualization field. There are two major categories in way to visualization. One is direct volume rendering, and the other is surface reconstruction. In case of three-dimensional space, the geometrical location of values can be directly represented by x , y , z coordination. Using direct volume rendering, each pixel is examined by light transfer function and their colors are calculated without extracting surfaces. However, deciding color of pixel using light transfer function is expensive computational process. In addition, it is hard to observe the inside structure of value distribution of the dataset. Furthermore, every time the view point and angle changes, colors of pixels should be calculated again.

The other way to visualization is that construct surface on the volume and render it using the method to render general three dimensional meshes. An isosurface, also called implicit surface, is one of the way to construct a surface on volumetric data. It is a surface which has constant value (e.g. density, intensity, temperature) in the three-dimensional scalar field. It is called an isocontour or contour lines in two-dimensional space. Given scalar field $f : \mathbb{R}^d \rightarrow \mathbb{R}$ and a constant value $\alpha \in \mathbb{R}$, the

isosurface on the isovalue α is defined by the level set $\{x | f(x) = \alpha\}$. But since it is just a point set, but not a perfect surface and data cannot be continuous in computer science, an isosurface has to be approximated using specific algorithms. Marching cubes algorithm is one of the widely used algorithm to approximate an isosurface on the volume dataset.

Figure 1 is example of each approach described in previous paragraphs. (a) is direct volume rendered image of laser intensity data, and (b) is its isosurface. Each surfaces of same color have same value.

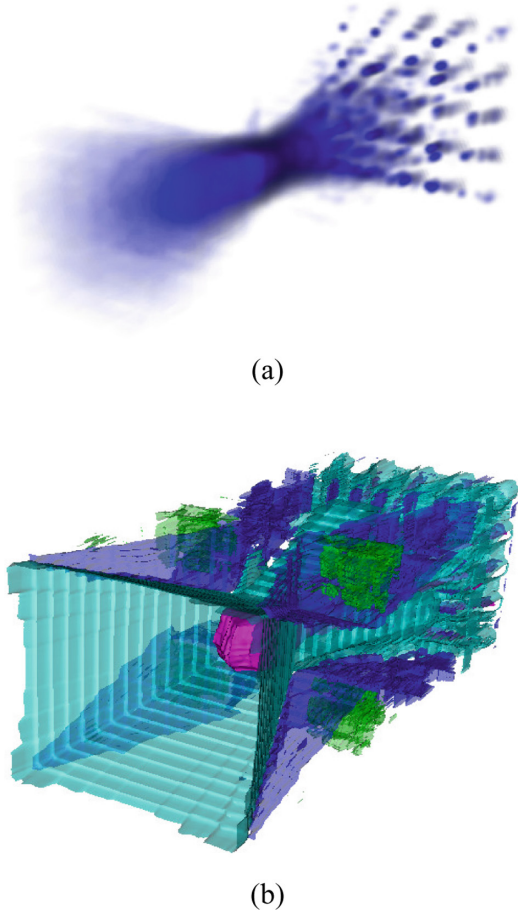


Fig. 1. (a) Image of laser intensity data using direct volume rendering and (b) Its isosurface.

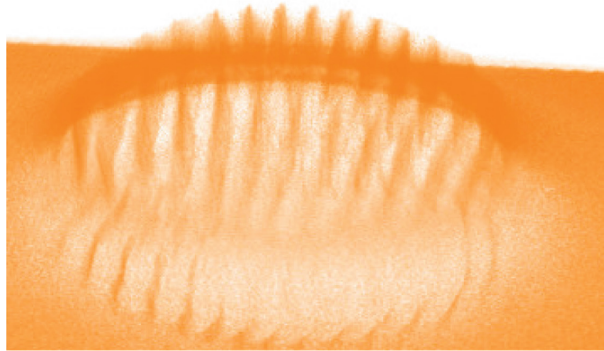
2 Problem Description

2.1 Properties of Volumetric Scattered Data

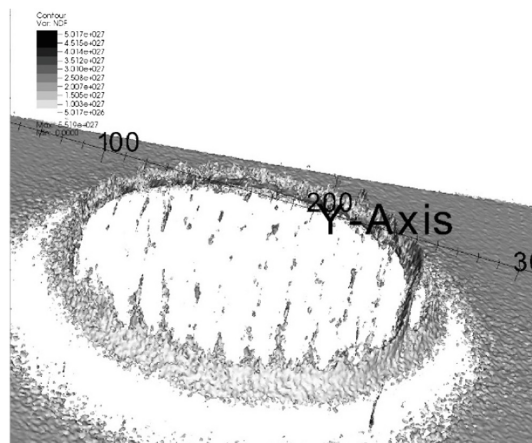
Scattered data has properties of unorganized samples. Each value is located at sparse and irregular position in the space. There are numerous practical examples of scattered data in computer science (e.g. computational fluid dynamics, terrain modeling, etc.). In tri-variate and scalar dependent value case, scattered data can be represented by $f(x_i, y_i, z_i) = F_i, i = (1, \dots, N)$ where $P_i = (x_i, y_i, z_i)$ are the independent variables and F_i is the dependent variable. In this case, the dataset is called volumetric scattered data.

2.2 Isosurface of Volumetric Scattered Data

Major problem of isosurface is that it is vulnerable to noise of data. Volumetric scattered data is discrete, and not a smooth function, therefore it does not guarantee connectivity of its generated surface. It is even not a surface in many case, because the



(a)



(b)

Fig. 2. (a) Volume rendered image of plasma charge density data and (b) Its isosurface.

level set of given function and an isovalue is not always a surface. For example, if f is constant function $f = \alpha$, its level set is all elements of function f 's domain when an isovalue is α , but empty set for all the other cases.

Another problem is that data must be organized in cube domain to extract an isosurface using Marching cubes algorithm that is a general method for extracting an isosurfaces of the three-dimensional data. Therefore, scattered data must be interpolated on all vertices of the cubes.

(a) in Fig. 2 is an example of volumetric scattered data and (b) is its isosurface constructed by using Marching cubes algorithm. Constructed surface consist of a lot of disconnected triangles. It is neither good from visual perspective nor easy to get information of geometrical distribution of the isovalue.

3 Related Work

3.1 Marching Cubes

Marching cubes, published in the 1987 SIGGRAPH proceedings by Lorensen and Cline, is famous algorithm for extracting an isosurface of the volumetric data. Marching cubes algorithm is a good choice to construct an isosurface of volumetric data of smooth function. This algorithm use the case table of triangle topology. Each voxel of volumetric data generates an triangle set lying with linearly interpolated point of the voxel's edge using the case table. Each process of generating triangles can be executed independently, so Marching cubes algorithm can be relatively easily accelerated by using parallel computing on multi-core CPU or GPGPU (General purpose graphic processor unit). Figure 3 is the case table of triangle topology, exclusive of symmetric cases.

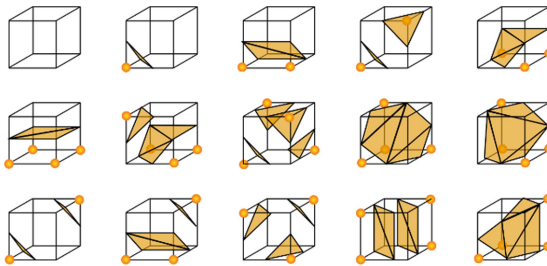


Fig. 3. 15 cases of triangle topology

3.2 Delaunay Triangulation and Voronoi Diagram

Delaunay triangulation is one of the triangulation method that maximize the minimum angle of the generated triangles. A Sliver triangle, which has a long and thin shape, may be inappropriate when used for interpolation or rasterization. In addition, Delaunay triangulation can be extended to three- or higher dimensional triangulation. In three-dimensional case, Delaunay triangulation generates tetrahedrals. Therefore,

Delaunay triangulation can be proper method of triangulation when deciding natural neighbors of the interpolation point. In Fig. 4, there is three ways to triangulate given point set $P = \{p_A, p_B, p_C, p_D, p_E\}$. It can be verified that all the angle of triangles is minimum by all the circumcircle of triangles does not contain any other points except for three points to consist of the triangle. In Fig. 4, (a) satisfies Delaunay triangulation properties, but the others do not.

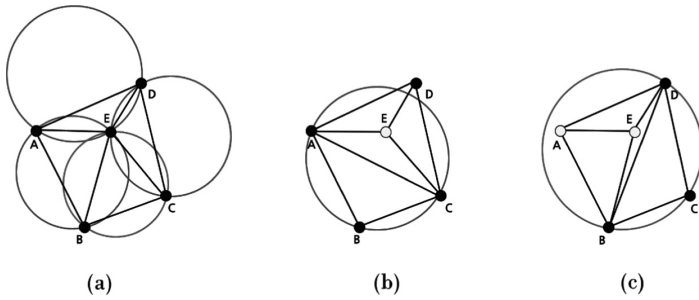


Fig. 4. Delaunay triangulation

Voronoi diagram is the subdivision of the space into Voronoi cells. Voronoi cells represent which point is closest to a position in the space. In other words, Voronoi cells of the point p_i is an influence of p_i in the space. Given point set P and its Delaunay triangulation $DT(P)$, Voronoi diagram of P , $VD(P)$, corresponds to the dual graph of the $DT(P)$. Voronoi diagram can be calculated by connecting the center of circumcircle which is generated by Delaunay triangulation. Figure 5 shows a relationship between Delaunay triangulation and Voronoi diagram.

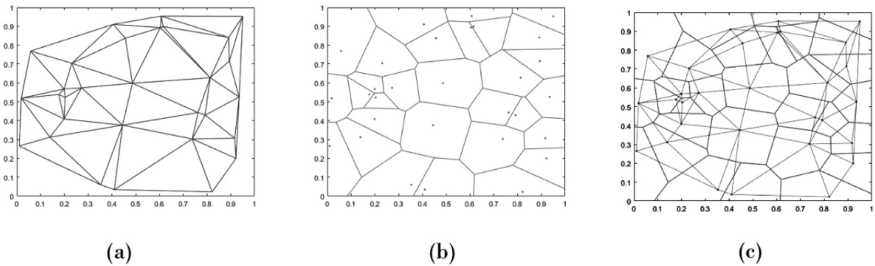


Fig. 5. Relationship between Delaunay triangulation and Voronoi diagram

3.3 Scattered Data Interpolation

Scattered data often approximated to mathematical modeling function to visualization the relationship of each scalar value in the dataset. Scattered data interpolation is the problem that find a function $F(x, y, z)$ which is its dependent variable is an approximated value of given scattered data $f(x_i, y_i, z_i) = F_i$ for a non-given point. There are

many options for interpolation method on scattered data. The simplest way of interpolation is Nearest neighbor interpolation. It does not consider a weight of given values but assign same value with its nearest point. This method closely connected to Voronoi diagram. All approximated points in a Voronoi cell has an identical value with the seed point of the cell in Nearest neighbor interpolation.

Another way of interpolation is a weighted average interpolation. A weight can be decided by various terms (e.g. distance based weight). The Shepard’s method is one of the basic weighted average interpolation method. It uses an inverse distanced weight for approximation. It can be write in following form where N is the number of points and $p_i, (i = 1, \dots, N)$ is given point of original function.

$$F(p) = \frac{\sum_{i=1}^N \frac{F_i}{\|p-p_i\|^2}}{\sum_{i=1}^N \frac{1}{\|p-p_i\|^2}}$$

Considering all the points in the given point set is inefficient for computation. So, many weighted average interpolation methods consider only a neighbor of an interpolated point. Neighbors can be chosen by whether p_i is within a certain distance from an interpolation point. Despite its simplicity, a distance based neighbor has some problems. If points in the data set are not uniformly distributed, distance based method may not work fine. In addition, it is hard to obtain continuous surface even if a data set is uniformly distributed.

4 Voronoi Natural Neighbor Interpolation

Voronoi natural neighbor interpolation offers a good interpolation result with acceptable computational resource. This method resolve two major problems of traditional weighted average interpolation methods; selection of neighbors, and continuousness of surface. It choose neighbor points in which the adjacent Voronoi cells. Given point set P , its Voronoi diagram $VD(P)$ and interpolation point x , insertion of x creates new Voronoi diagram $VD(P \cup \{x\})$. Created Voronoi cell \mathcal{V}_x alters the original Voronoi cells \mathcal{V}_{p_i} which is adjacent to \mathcal{V}_x . In Natural neighbor coordinates, new point x represented by weighted summation of its neighbors. A weight can be written following form.

$$w_i = \frac{Area(\mathcal{V}_{p_i} \cap \mathcal{V}_x)}{Area(\mathcal{V}_x)}$$

Where $Area(\mathcal{V}_{p_i})$ represents the area of the Voronoi cell \mathcal{V}_{p_i} . The term $\mathcal{V}_{p_i} \cap \mathcal{V}_x$ is a stolen area caused by insertion of x . Voronoi natural neighbor interpolation use this weight w_i for weighted interpolation. Interpolated function $F(p)$ can be represented by

this form where N is the number of neighbors. Interpolated function $F(p)$ is continuous everywhere except original point.

$$F(p) = \sum_{i=1}^N w_i F_i$$

(a) in Fig. 6 is the original Voronoi diagram, (b) is newly created Voronoi diagram by insertion of x , and (c) is an overlapped image of previous two images showing weight w_i , which is stolen area of the original Voronoi cell.

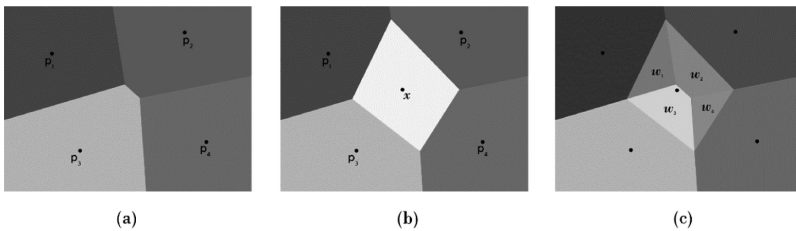


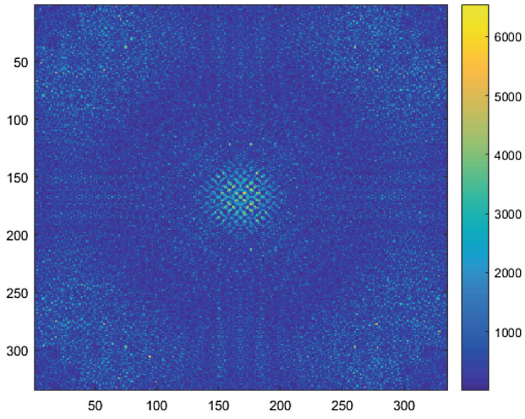
Fig. 6. Voronoi diagram and Natural neighbors interpolation.

5 Results

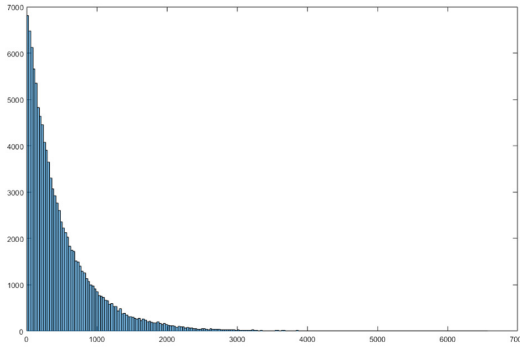
Experiment was performed by comparing isocontours of three different datasets. First is the original dataset of laser intensity data. Second is the dataset smoothed using Gaussian filter. Last one is the dataset which is down-sampled from original image and approximated to original resolution by using Voronoi natural neighbor interpolation.

The isocontour of original dataset shows many islands, so it is relatively difficult to know intuitively geometrical distribution of isovalues. The second one, Gaussian filtered dataset, shows less islands than previous one, but the range of the value is significantly changed when averaging the values out. In fact, the original dataset's range of value is $4.5e-10$ to $6.5501e+03$, but the second one is 52.9433 to $2.3670e+03$. On the other hand, the isocontour of interpolated dataset shows less islands and modulation does not occur during interpolation process except during down-sampling process.

Figures 7 and 8 are a visualized image of the original dataset and its isocontour respectively. Figures 9 and 10 are images of the Gaussian filtered dataset and Figs. 11 and 12 are images of the interpolated dataset's respectively.



(a)



(b)

Fig. 7. (a) Original dataset and (b) Its histogram.

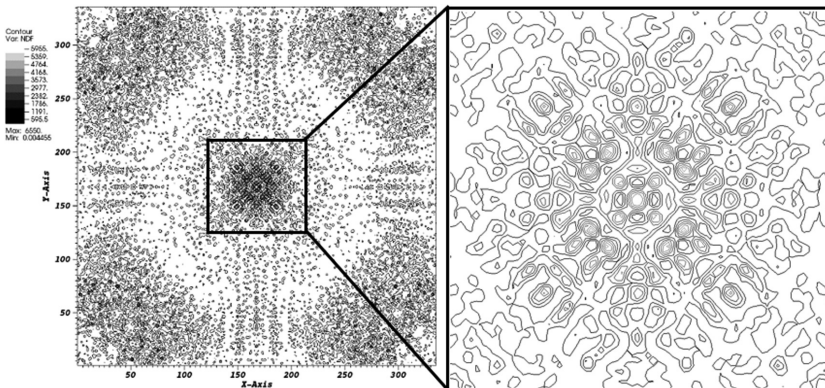
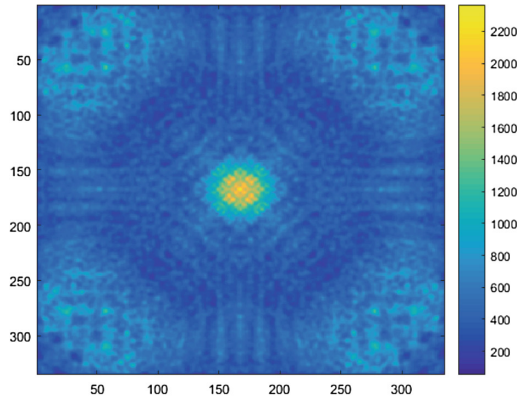
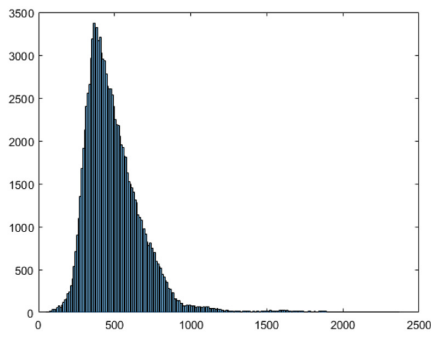


Fig. 8. Isocontour of the original dataset



(a)



(b)

Fig. 9. (a) Gaussian filtered dataset and (b) Its histogram.

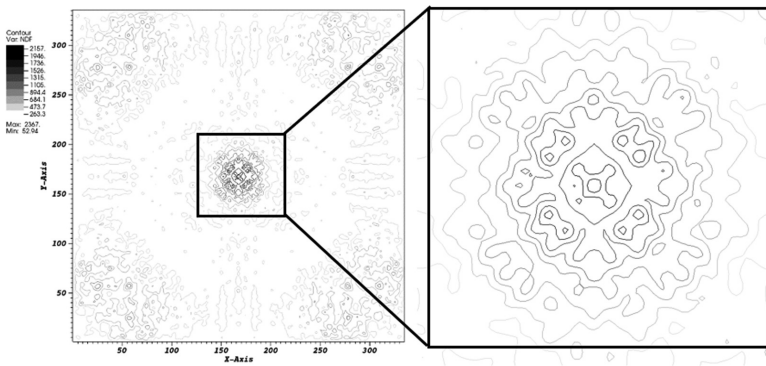


Fig. 10. Isocontour of the smoothed Gaussian filtered dataset.

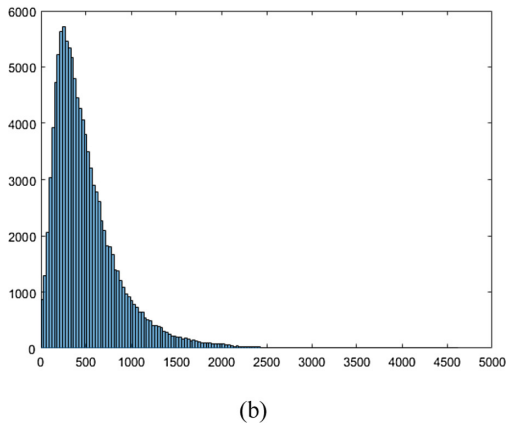
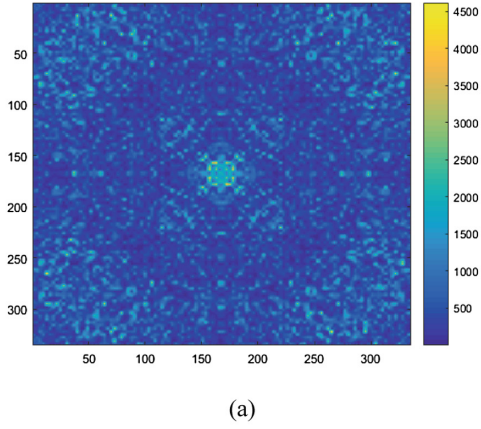


Fig. 11. (a) Interpolated down-sampled dataset and (b) Its histogram.

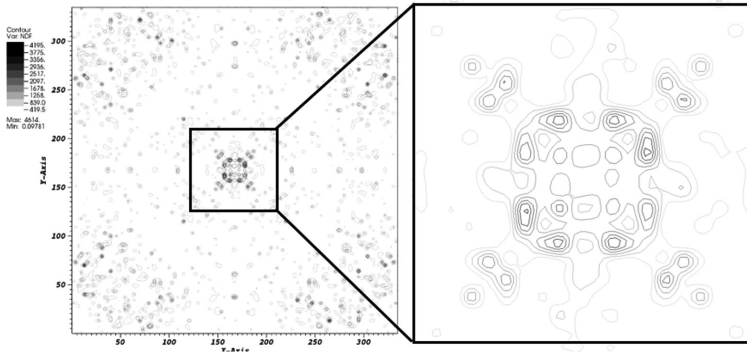


Fig. 12. Isocontour of the interpolated dataset.

6 Conclusions

Scattered data is everywhere in the computer science field. Scattered data is hard to visualize or be analyzed in itself. So reconstruct a surface of scattered data is important issue in computational geometry. This paper showed how to get approximate isosurface or isocontour using interpolation, and described advantages of Voronoi natural neighbor interpolation against other interpolation method. In addition, It showed that problem of using smoothing filter to get less noise isosurface and showed that proposed method solves this problem.

In this paper, uniform down-sampling was used, but it may be valuable to compare the result using other method. In further research, I will compare the result using feature based down-sampling method (like local maximum and curvature) and find proper way to down-sample the original data.

Acknowledgments. This work was supported by the Industrial Strategic technology development program, 10048964, Development of 125 J-Hz laser system for laser peering funded by Ministry of Trade, Industry & Energy (MI, republic of Korea).

References

1. Rübél, O., Geddes, C.G.R., Chen, M., Cormier-Michel, E., Bethel, E.W.: Feature-based analysis of plasma-based particle acceleration data. *IEEE Trans. Vis. Comput. Graph.* **20**(2), 196–210 (2014)
2. Wenger, R.: *Isosurfaces*, pp. 1–16. CRC Press, Boca Raton (2013)
3. Telea, A.C.: *Data Visualization: Principles and Practice* 2nd edn. CRC Press, Boca Raton (2015)
4. Nielson, G.M.: Scattered data modeling. *IEEE Comput. Graph. Appl.* **13**, 60–70 (1993)
5. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. *Comput. Graph.* **21**(4), 163–169 (1987)
6. Dyken, C., Ziegler, G., Theobalt, C., Seidel, H.-P.: High-speed Marching Cubes using HistoPyramids. *Comput. Graph. Forum* **27**(8), 2028–2039 (2008)
7. Lee, K.: Visualization of trivariate scattered data interpolation. *J. Korea Comput. Graph. Soc.* **2**(2), 11–20 (1996)
8. Sibson, R.: A brief description of natural neighbour interpolation. In: Barnett, V. (ed.) *Interpreting Multivariate Data*, pp. 21–36. Wiley, New York (1981)
9. Boissonnat, J.D., Cazals, F.: Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Comput. Geometry* **22**, 185–203 (2002)
10. Ledoux, H., Gold, C.: An efficient natural neighbour interpolation algorithm for geoscientific modelling. In: *Developments in Spatial Data Handling*, pp. 91–108. Springer, Heidelberg (2005). https://doi.org/10.1007/3-540-26772-7_8