

**Barbara PUTZ, Paweł MATYSKIEŁA**  
Warsaw University of Technology, Warsaw

## **INTERPOLATION OF IRREGULAR SETS OF POINTS BY SUBDIVISION SURFACES IN NUTRIA MODELLER**

### **Key words**

Meshes, interpolation, subdivision, triangulation, Butterfly, refinement.

### **Summary**

The paper presents the solution of the problem of interpolation the irregular set of 3D data points by a surface represented as the mesh of triangular faces. The surface should be smooth with some tolerance given. The task has been solved in a non-trivial way, with the use of known algorithms, in two steps. First the input points are triangulated by the Ball Pivoting Algorithm, originally designed for scanned point clouds. Then the modified Butterfly subdivision algorithm is performed. The implementation is dedicated for the NuTria modeller, destined for bent metal plates.

### **Introduction**

During the designing of the 3D free-form surface modeller destined for small and medium-sized enterprises [5] we have got the task to find a smooth surface interpolating given unorganised 3D data points set  $S \in \mathcal{R}^3$ . These input points form the shapes typical for bent metal plates. The output surface should be defined as a mesh of triangles, being of class  $C^1$  with some acceptable tolerance. Below we present the steps which we have performed in a some non-typical way to resolve the problem.

## 1. Triangulation by the Ball Pivoting Algorithm

A finite point set  $S \in \mathcal{R}^2$  defines Delaunay triangulation in 2D space, which is unique and decomposes the convex hull of  $S$  into triangles. This subdivision has the property that apart from the vertices, there are no other points on or inside the circumscribed circle of any triangle (Fig. 1a):

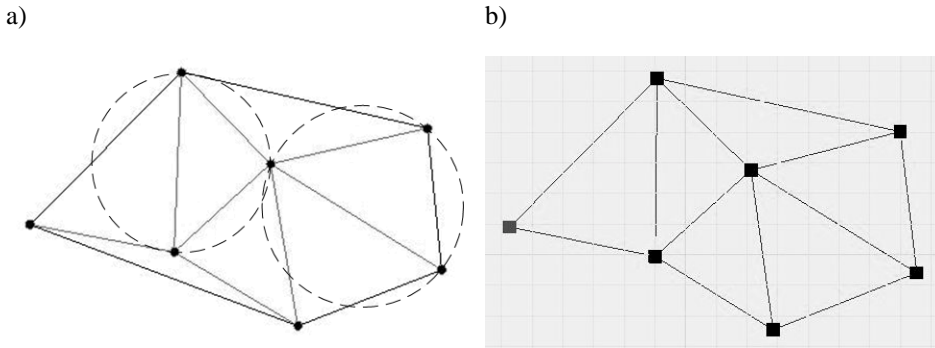


Fig. 1. a) 2D Delaunay triangulation; b) 2D triangulation in NuTria with BPA

Analogously, Delaunay triangulation defined for the finite point set  $S \in \mathcal{R}^3$  is unique and decomposes the convex hull of  $S$  into tetrahedra. This subdivision has the property that apart from the vertices, there are no other points on or inside the circumscribed sphere of any tetrahedron.

The Ball-Pivoting Algorithm (BPA) [2], closely related to alpha-shapes [4], used for surface reconstruction, is a subset of 2-faces of 3D Delaunay triangulation of  $S$ . Thus, contrary to Delaunay triangulation, it can be concave (Fig. 1b).

The main idea of the Ball-Pivoting Algorithm uses a ball of user-specified radius  $R$  ( $R$ -ball). Three points form a triangle, if  $R$ -ball touches them without contacting any other point. Starting with a seed triangle, the ball pivots around an edge (formed by two of these initial points), until it touches another point, forming another triangle (Fig. 2a). This way  $R$ -ball pivots around each edge of the current mesh boundary, forming, during such a "walk", the interpolating mesh of triangles. If the point density is too low in respect to radius  $R$ , some of the edges will not be created, leaving holes (Fig. 2b). If the radius of local curvature is less than  $R$ , some of the data points will not be touched; the pivoting ball has multiple contacts with the surface in this case (Fig. 2c).

The algorithm implemented in NuTria modeller does not use the spatial query with a regular grid of voxels as the the original BPA, because there is no a huge amount of points to triangulate, as in the case of point cloud acquired with multiple range scans. Besides this simplifications, some other modifications of the original BPA have been introduced in NuTria, as presented below.

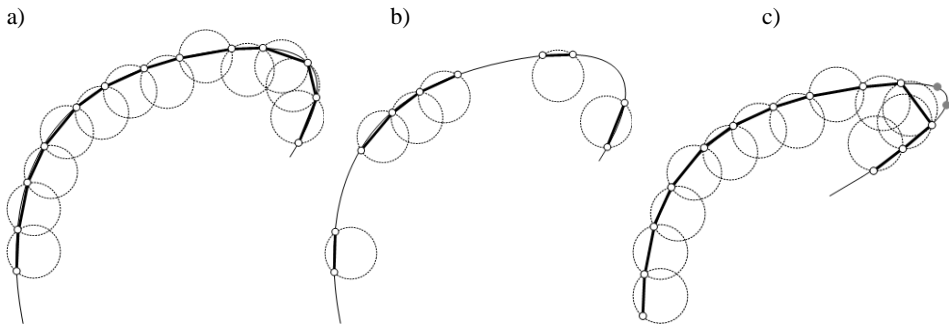


Fig. 2. The 2D-idea of the Ball-Pivoting Algorithm

The triangulation process in NuTria starts from the selection of the seed triangle. Its base is the shortest segment between two data points, selected by searching process. The third point of the first triangle is created by pivoting of the ball around this segment – until during this motion the ball hits some point.

Choosing the radius of the pivoting ball in the original BPA is easy with knowledge of the sampling density characteristics of the scanner. In the case of uneven sampling the BPA can be applied multiple times, with increasing ball radii. In NuTria algorithm the radius of the pivoting ball is calculated for every new triangle on the base of the length of the segment being the pivot axis (the value of the ball radius is slightly bigger than half of this length).

The searching of the third point of the wall is performed always in the range of the angle given by the user (the *AngleRange* value). This range is placed symmetrically in relation to the plane of the existing triangle including the pivot axis (Fig. 3). In the case of the creation of the first triangle – starting with the triangulation process – we assume that its normal versor is close to *Oz* axis as much as possible.

In the first step of searching the third point of every next triangle we find the radius  $r_o$  of the circular trajectory of the centre of the pivoting  $R$ -ball (Fig. 3):

$$r_o = \sqrt{R^2 - \frac{|\mathbf{AB}|^2}{4}}$$

Next we select for further analyses those points whose distance from the middle point  $\mathbf{M}$  of the edge  $\mathbf{AB}$ , being the pivot axis, is less than diameter of the pivoting ball ( $<2R$ ). Then for every selected point  $\mathbf{P}$  we find the radius  $r_s$  of the circle, being the intersection of the plane  $\pi$ , perpendicular to the pivot axis  $\mathbf{AB}$  and crossing it in the middle point  $\mathbf{M}$ , and the sphere of radius  $R$  centred in the analysed point  $\mathbf{P}$  (if the distance from  $\mathbf{P}$  to the plane  $\pi$  is less than  $R$ ):

$$r_s = \sqrt{R^2 - |\overline{PP_s}|^2}$$

where  $P_s$  – the point of the plane  $\pi$ , closest to the point  $P$ .

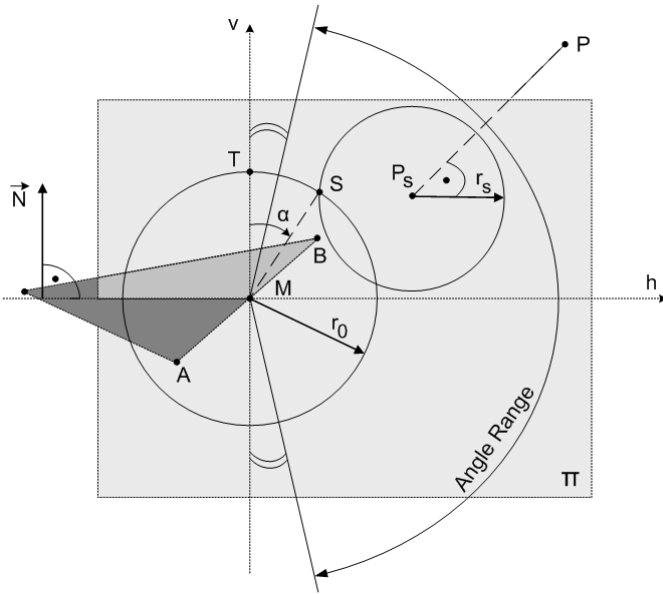


Fig. 3. All the points of the circle with radius  $r_s$  are lying at distance  $R$  (the radius of the pivoting ball) from the data point  $P$ . The point  $P_s$  must lie in the region described by the Angle Range. The circle of radius  $r_0$  is the trajectory of the centre of the pivoting ball. If both circles intersect (the angle  $\alpha$  describes the position of its first point of intersection  $S$ ), then the pivoting ball touches point  $P$

Analysing the points of intersection of circular trajectory with the circle of radius  $r_s$  for all selected points  $P$ , we can find among these selected data points the one which will be touched by the pivoting ball first when its centre moves through the plane  $\pi$  around point  $M$  (in clockwise direction if we refer to Fig. 3).

If we can't find any point  $P$  meeting all conditions, the radius of the pivoting ball is increased and the process is repeated up to finding the third point of the triangle. Then, if both edges of the new triangle haven't been used yet, the new pivot axis is chosen from those lying on the left side of the previous pivoting edge.

If the pivoting ball can't touch during such steps any other point (the number of admissible increasing steps is given by the user), the segment used as the pivot axis is considered as the boundary edge of the surface, and we go to the next segment on the left.

Walking this way, the pivoting ball connects all the data points into regular mesh forming one side of the surface, bounded by the edges. This is true in the case of regularly sampled data points. Sometimes however we can have missing points, which are not touched by the ball. This is the case when some single points are apart from all others (Fig. 4a) and the interpolated surface would be too bent. It can be the effect of a user mistake during manual data entry. Thus the algorithm can help to detect and visualise such errors in data. The user can then edit such points and call the process of retriangulation (Fig. 4b). Instead – if he decides he wants to include to surface this outlying point – he can allow the greater range of increasing the radius of the pivot ball.

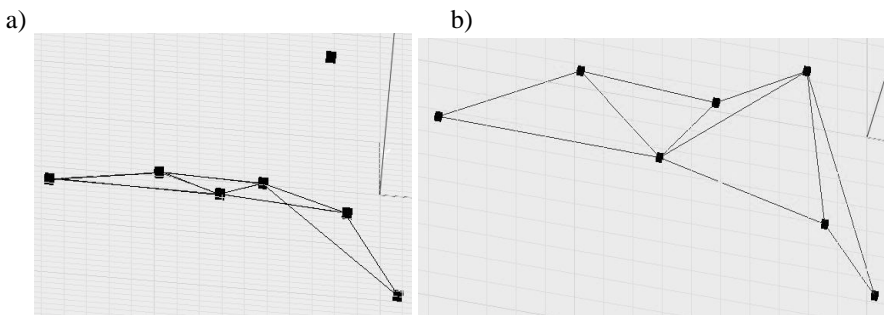


Fig. 4. a) Some too far outlying data points can be missed during the triangulation by the BPA in NuTria; b) Retriangulation after moving the lonely point closer to other points

To avoid ambiguities and crossing edges in the case of user mistakes in data provided, some additional assumptions have been introduced into the process of triangulation. The plane  $xOy$  is used as a reference plane together with the assumption that the surface, which we expect to obtain by triangulation, can be described in a simple form  $z = f(x,y)$ . Due to this limitation the pivoting ball is creating the surface lying below this ball. This limitation is also closely connected to the manner the user can provide the data, i.e. they are provided as the points extruded from the plane  $xOy$  (if the reference plane  $xOy$  is not useful, NuTria offers the possibility to change the coordinate system). Thus if two data points have almost the same values of the  $x$  and  $y$  coordinates, so as they differ only by  $z$ -coordinates, one of them will not be used as a surface mesh point.

The user can additionally and optionally force – by the algorithm – the linking to the mesh of some missing, distant data points. In such case of “manual” control, the mesh obtained does not retain properties to be a subset of Delaunay triangulation, however it fulfils all necessary properties of the mesh describing the open surface. Thus such option can be interesting for the user. This extra linking is performed by connecting the missed point to the vertices of the

nearest face or of the face it is projected on in  $O_z$  direction – with subsequent retriangulation in some surroundings.

## 2. Subdivision by the modified Butterfly algorithm

The rough mesh obtained during triangulation needs to be refined with the assumption that the input, original vertices of the mesh cannot be changed. To fulfil these requirements, the idea of interpolating subdivision scheme proved to be a good solution. The local and simple Butterfly scheme defined on a triangular grid [6, 8] has been chosen for splitting each edge of triangular face and reconnecting:

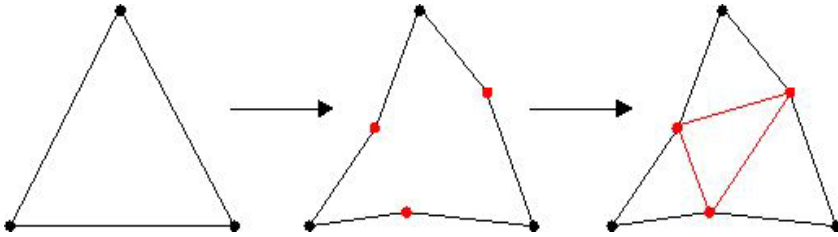


Fig. 5. The idea of triangular mesh refinement

The new vertices in the Butterfly scheme are lying near the midpoints of the original edges and for every edge they are defined as a linear weighted combination of the vertices of two triangles adjacent to the edge and of four triangles adjacent to them. Thus every triangular face is divided into four new faces, but in every step of the subdivision the existing vertices are not changed, so the original vertices of the mesh are retained (Fig. 5).

To obtain better results of interpolation and avoid undesirable creases, the modified subdivision Butterfly scheme [8] has been used in which the weights associated with the vertices are defined as a function of parameter  $w$ . Additionally, the stencil of neighbouring vertices in this modified scheme is not 8-point, as in the original Butterfly scheme, but it is a 10-point scheme in the regular case, when every vertex has valence equal to 6, i.e. there are 6 edges incident to the vertex (Fig. 6). In the implementation of the modified algorithm in NuTria the value  $w=0$  has been used, as it has been suggested by the authors of this algorithm, and it gives good results.

In the case when the edge has one extraordinary vertex at the end, i.e. we have  $N \neq 6$  edges incident to this vertex, the weights of the vertices in the modified Butterfly algorithm depend on the value of  $N$ , as presented in Fig. 7.

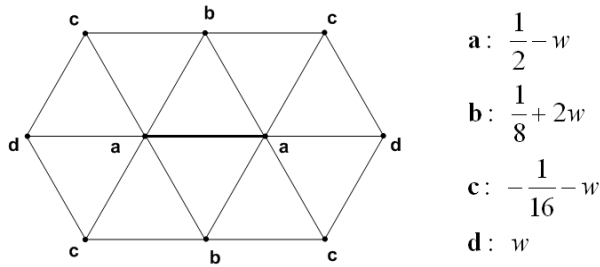


Fig. 6. The modified Butterfly algorithm in the regular case for the central edge, when it connects two vertices of valence 6. To split the marked edge and calculate the new vertex, we need the different weights for different types of vertices (a, b, c, d); these weights are function of parameter  $w$ , as presented on the right. In the case  $w=0$  the algorithm reduces to the original Butterfly algorithm

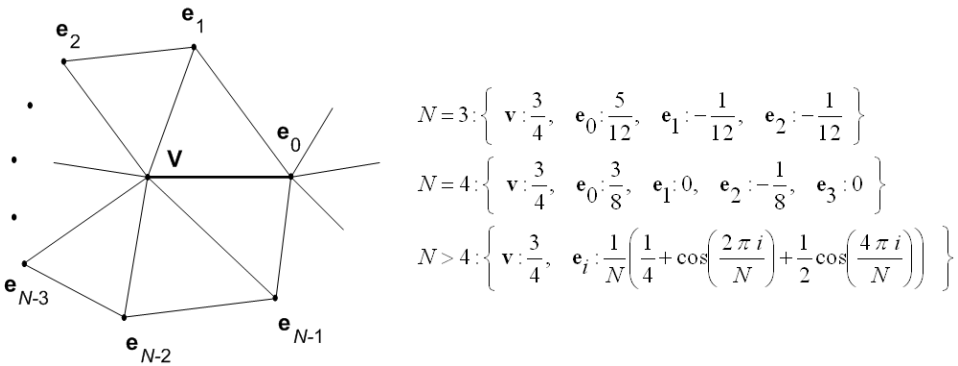


Fig. 7. The modified Butterfly algorithm in the case when the edge to be splitted (marked) connects 6-vertex (vertex of valence 6)  $e_0$  and  $N$ -vertex ( $N \neq 6$ )  $v$ , for which its neighbours  $e_1, \dots, e_{N-1}$ , with weights as presented on the right, are used to calculate a new point

If the edge connects two extraordinary vertices, the new point of the splitted edge is calculated as the average of the points computed using the above scheme relevant for single extraordinary point.

To calculate a new vertex for boundary edges we need only 4 vertices, with weights as below (Fig. 8):

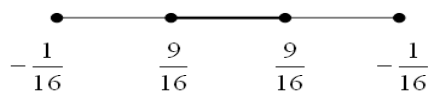


Fig. 8. The modified Butterfly algorithm in the case of splitting boundary edge (marked)

After every step of refinement defined as below, every triangular face is divided into four new triangles. The process can be continued using the following modes:

- step by step – until user decides about its finish,
- continuously – until the difference between the edge before splitting and the new point of the splitted edge is below the value given by the user,
- continuously – until the difference between two subsequent areas of refinement surfaces is below the value given by the user (in %).

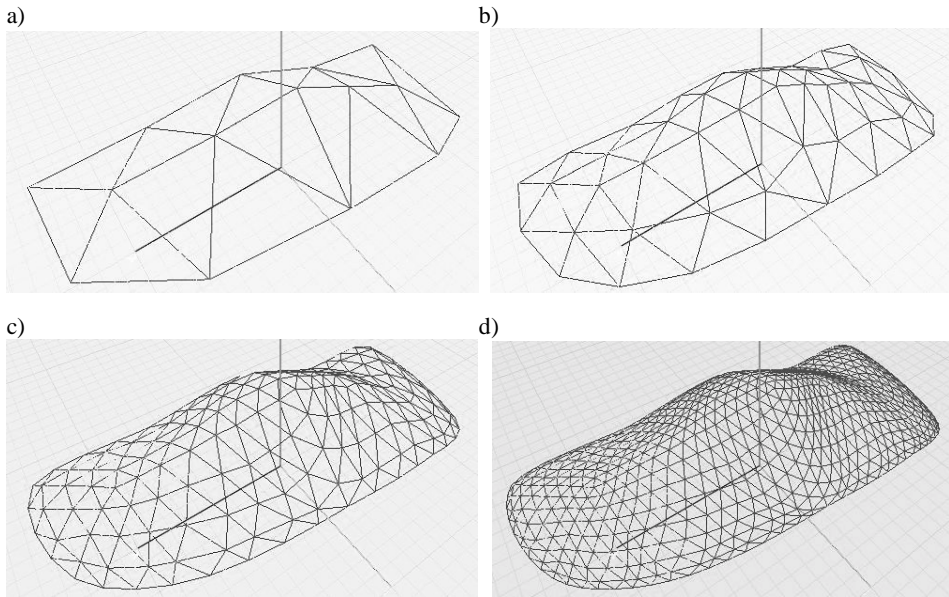


Fig. 9. The mesh after triangulation (a) and the three next steps of its refinement by the modified Butterfly algorithm; in every step the initial mesh points (a) are interpolated

### 3. Halfedge data structure

For encoding the mesh with efficient operations of subdivision enabled, a halfedge data structure [3] has been used. This type of data structure does not require any search operations during traversals.

In halfedge data structure each edge of the mesh is split into two twin halfedges, oriented in the opposite directions (Fig. 10). Each halfedge stores a reference to:

- the vertex it points to (the incident vertex),
- the face on its left,
- the next halfedge of the face (in counter-clockwise direction),
- the opposite halfedge,
- the previous halfedge of the face.



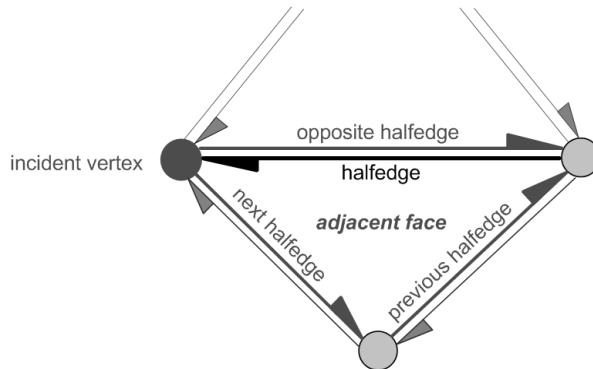


Fig. 10. The halfedge data structure

It is not the minimal halfedge structure (the minimal one maintains only halfedges with next and opposite pointers), but it is useful to get better performance and conveniently used in applications. Some of references may be null in the case of boundary edges.

The halfedge data structure, implemented in NuTria as dynamic list with the use of pointers, proved to be a good, efficient solution in the process of triangulation and subdivision.

## Conclusions

The non-trivial use of the Ball Pivoting Algorithm has been presented, followed by the modified Butterfly scheme. If the initial mesh is sampled from a smooth surface, this coupled scheme of interpolation performs quite well. If no, the scheme is sensitive to the errors and irregularities of the input mesh and user interaction is necessary. The use of adaptive subdivision scheme [1] and nonlinear subdivision schemes [7] can be considered as the alternatives to modified Butterfly scheme.

## References

1. Amresh A., Farin G., Razdan A.: Adaptive Subdivision Schemes for Triangular Meshes, 2001. <http://citeseer.ist.psu.edu/558448.html>
2. Bernardini F., Mittleman J., Rushmeier H., Silva C., Taubin G.: The Ball-Pivoting Algorithm for Surface Reconstruction. *IEEE Transaction on Visualization and Computer Graphics*, Vol. 5, no. 4, pp. 349–359, 1999.
3. Campagna S., Kobbelt L., Seidel H.-P.: Directed Edges – A Scalable Representation For Triangle Meshes. *ACM Journal of Graphics Tools* 3 (4), 1998.

4. Edelsbrunner H., Mücke E.P.: Three-dimensional alpha shapes. *ACM Transactions on Graphics*, vol.13, no 1, pp. 43–72, 1994.
5. Putz B., Przybylski M., Kaźmierski S., Matyskieła P., Zagroba K.: NuTria – system of modelling of NURBS and triangulated surfaces. *Maintenance Problems*, Institute for Sustainable Technologies-National Research Institute in Radom.
6. Warren J., Weimer H.: *Subdivision Methods for Geometric Design. A Constructive Approach*. Academic Press 2002.
7. Yang X.: Surface Interpolation of meshes by geometric subdivision. *Computer-Aided Design*, vol. 37, pp. 497–508, 2005.
8. Zorin D., Schröder P., Sweldens W.: Interpolating subdivision meshes with arbitrary topology. *ACM SIGGRAPH Proceedings*, pp. 189–192, 1996.

Reviewer:

**Wojciech CHOLEWA**

*Scientific work financed by the Ministry of Science and Higher Education and carried out within the Multi-Year Programme “Development of innovative systems of manufacturing and maintenance 2004-2008”.*

## **Interpolacja nieregularnych zbiorów punktów za pomocą powierzchni subdivision w modelerze NuTria**

### **Słowa kluczowe**

Interpolacja, triangulacja, Butterfly, siatki trójkątów, wielokrotne podziały.

### **Streszczenie**

W artykule przedstawiono rozwiązanie problemu interpolacji nieregularnych zbiorów punktów za pomocą siatek trójkątów tworzących powierzchnię o gładkości zbliżonej do  $C^1$  z zadaną tolerancją. Zadanie zostało wykonane w niezbyt typowy sposób z wykorzystaniem znanych algorytmów: algorytmu triangulacji metodą przetaczającej się kuli i algorytmu iteracyjnych podziałów zmodyfikowaną metodą Butterfly. Wynikiem implementacji jest moduł interpolacji zadanego zbioru punktów w modelerze NuTria, przeznaczonym do modelowania blach giętych w MSP.