Implicit Surface Reconstruction and uniform Sampling by *Striped* Marching Triangles

Víctor Leborán and Xose M. Pardo

Artificial Vision Group. Department of Electronics and Computer Science.
University of Santiago de Compostela, 15782. Santiago de Compostela. Spain.
Email: nelbla@usc.es, pardo@dec.usc.es

Abstract

The synthesis of computer models from real objects is a usual procedure in medical image processing and reverse engineering. In both cases, the main issues are to reach a topologically consistent and geometrically precise object reconstruction. Implicit surfaces provide a solution to this problem. They define a smooth surface surrounding objects which must be polygonized for an efficient visualization. Due to the increasing size of the application generated data, it becomes necessary to use precise and efficient methods. This work presents a modified method of implicit surface polygonization based on "Marching Triangles". This modification assures the topological consistency with the initial dataset during the progressive reconstruction of the surface, avoiding overlapping triangles. In addition, a comparison of our method with the marching cubes and the adaptive skeleton climbing algorithms is provided.

Keywords:
Implicit surfaces, Euler operators, Delaunay triangulation, Marching Triangles.

1. Introduction

Poligonization based on triangles is a common visualization method for implicit models, such as those generated from bones images in computed tomography. Parametric surfaces are assumed as the main tool for the generation of smooth surfaces in most of the interactive modelling environments. Nevertheless, implicit surfaces have got several advantages over parametric ones like the possibility of being combined and deformed [19]. They make easier the smooth surface construction with any topology and geometry.

Implicit surfaces can be obtained from discrete or continuous datasets. For example, 3D medical image lies in the discrete case, whereas the algebraic surfaces, obtained by combination of implicit algebraic equations, fall into the continuous case. Implicit surfaces are typically defined by starting with simple building block functions and by creating new functions using operators like sum, min or max. If the building blocks are spherical Gaussian functions the surfaces are called blobbies, soft objects or metaballs [22].

An implicit surface is defined as the set of points P satisfying the implicit equation $f(P) = c$ for some function f [5,17]. Methods that construct the approximation to this surface are known as implicit surface tilers or iso-surface generation algorithms when applied to varying values of the value c. Mainly there are two kinds of tiling methods, volume based and surface based methods.

Volume based methods partition the space in non-overlapping polyhedral cells. They use a volumetric decomposition of the space to polygonize the implicit surface. The goal of the tiler is to locate intersections of the cell edges with the implicit surface to obtain an approximation of the surface inside the unit cell. The result will be represented by one or more polygons [17].

As the function evaluation for every point is a high time-consuming task, a valuable alternative approach consists of starting with an initial number of cells on the surface and propagating them in a selective way. This method is known as numerical continuation [2] and greatly decreases the number of evaluated cells.

An alternative to the volume based implicit surface tiling approach are the surface based methods. Marching triangles is a surface based method which tiles the surface according to local geometry and topology [6]. The topology of an implicit surface refers to the number of disjoint components together with the genus of each component. A topologically correct polygonization must share the same number of disjoint components, each component having the same genus as its corresponding component on the implicit surface [20].

There are also other kinds of methods for visualizing and controlling an implicit surface, some of them are related to the surface sampling. These techniques allow direct manipulation of the implicit volume and offer a preview of the final shape of the modified object. An example of this kind of methods can be found in the work of Witkin and Heckbert [22]. An iterative fissioning approach is combined with local repulsion forces to reach a surface with the desired sampling density.

2. Applications and related work

The implicit surface formulation defines a smooth surface around the object. Among other uses, this method can be employed in: efficient collision detection, exact contact modelling, volume preservation [3], modelling of soft substances that can be separated or melted, animation of elastic bodies or simple characters made of articulated skeletons coated with an implicit specified volume [12]. Recent applications of implicit surfaces can be found in cloth simulation within the scope of the fashion [4] and in the interactive animation of deformable tissues for surgery training [11].

The major drawbacks of the implicit surfaces compared to the parametric ones are the difficulty in achieving interactive visualizations, and the complexity of mapping 2D textures on these surfaces [10,21,23].

Representations based on particle systems have been proposed to tackle real time modelling. One of their shortcomings is that the observer must infer the shape of the surface from the particles positions and orientations. Particles polygonization offers an improved visual aspect at the expense of real time [22].

To deal with real time problem in the scope of implicit surfaces many solutions have been proposed, for instance the use of critical points. In these points the changes of the isosurface generation functions will produce topological changes on the surface [19, 20]. The algorithm only has to track those points to detect the changes in topology. This idea has been implemented in the extension of the shrinkwrap algorithm for isosurfaces with arbitrary topology introduced by Bottino et al. [7].

Marching triangles is a surface based method that offers an accurate and efficient representation of the implicit surface[14]. The adaptive modification proposed by Akkouche and Galin [1] solves the overlapping triangles problem and improves the original method allowing a different size of the triangles depending on the surface curvature.

Our proposal yields another surface based solution to the tiling procedure that also solves the aforementioned problem. It acts in two stages. At the end of the fist one we obtain a mesh of points that is a good uniform sampling of the final surface and an incomplete triangulation of the surface, being a partial view of the final shape. During the second stage, the surface of the entire object is reconstructed by filling the cracks generated from the first stage.

The remainder of this paper is organised as follows. We begin in Section 3 with a mathematical background as well as a brief description of the marching triangles algorithm [13,14]. Section 4 explains our proposal in detail. After describing the results and the limitations in Section 5 we conclude with a discussion of future work in section 6.

3. Theoretical basis

We propose an algorithm based on the 3D Delaunay triangulation constraint (see section 3.1), as well as on marching triangles algorithm (see section 3.3), proposed by Hilton and Illingworth [14].

Our algorithm was implemented with CGAL libraries that use local Euler operators for surface reconstruction (see section 3.2). As a result we obtain a polyhedron like the one defined by CGAL libraries [9]. The underlying data structure is based on the halfedge data structure. This construction makes easier the triangles subdivision of the final result of our algorithm to reach a smooth and accurate surface.

In the next sections we will use the following notation introduced by Akkouche and Galin [1]. A triangle $T(x_i,x_j,x_k)$ refers to a triangle with vertices x_i, x_j and x_k. The orientation of a triangle $T(x_i,x_j,x_k)$ is defined by its normal n_t. With c_t we refer to the circumcenter of the triangle T in the plane of the triangle. And x_i,x_j,x_k are vectors that represent the displacement from the origin.

3.1. Delaunay constraint

The Delaunay constraint provides the mathematical basis of the implemented method. Boissonnat has demonstrated the following property [6]:

> Given a set of points $X=\{x_0,\ldots,x_n\}$ in $\mathbb{R}^3$. Whenever the set of points X lie on a manifold surface, the Delaunay triangulation is characterized by the condition that it is composed of triangles $T(x_i,x_j,x_k)$ such that there exists a sphere passing through the vertices of $T(x_i,x_j,x_k)$ that does not contain any other point of the set X.

The above property can be used as the basis for deriving a local constraint of the triangulation [13, 14].

3D Delaunay Surface Constraint: A triangle $T(x_i,x_j,x_k)$, may only be added to the mesh boundary at edge $e(x_i,x_j)$ if no part of the existing model M' with the same surface orientation is inside the sphere passing through the triangle vertex x about the centre, c_t. Surface points of the same orientation are defined by a positive normal product [13,14].

The orientation criterion allows the polygonization of implicit surfaces with thin folding sections. This surface constraint is a somewhat relaxed Delaunay constraint, as stated in the work of Akkouche and Galin[1].

3.2. Euler operators and polyhedral surfaces.

This section reviews the basic theory of Euler operators (see [8] for further details). These operators and the concept of polyhedral surface are used in the CGAL libraries. Our implementation of the modified algorithm calls many functions of these libraries. Let us start with some definitions:

Definition 1. If we consider a modeling space based on properties of the boundary and not on the volume of solid objects, the boundary of a solid consists of a collection of faces that form a closed surface.

Definition 2. A topological transformation is defined as a continuous transformation that has a continuous inverse transformation. Two sets are topologically equivalent if there is a topological transformation between them.

Definition 3. A cell is any figure topologically equivalent to a closed disk.

Definition 4.. A polygon is a cell with a finite number of points on the boundary chosen as vertices. Sections of boundary between vertices are called edges.

Definition 5. A polyhedron is a complex that is topologically equivalent to a sphere and the constituting cells are called faces.

We can consider a special topology on plane figures where a point p on a labelled edge α has a neighbourhood consisting of the union of two half-disks around symmetrical points on identical labelled edges. This construction introduces the concept of plane models (PM) and allows the study of their topological properties. Figure 1 shows an example of a cylinder and a sphere and their PM.

Natural operations in contour representations come from considering well-formed PM and their properties. It becomes necessary to find a small set of correct PM operations that are enough to describe all PM of physical importance [8,16].

Topological properties do not depend on how the surface is divided to form a plane model. We consider PM with the same topological properties to represent the same surface (in a topological sense). Operations that move one surface from one subdivision to another do not change topological properties. Two such operations are polygon cutting and pasting and another two can be derived by considering the dual of a PM. These operations are vertex splitting and joining.

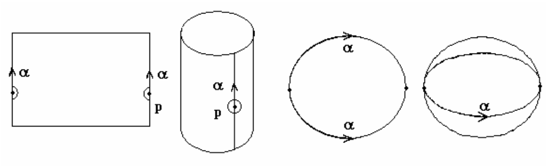


Figure 1. *Example of a plane model, a cylinder (left) obtained by gluing together a pair of opposite edges, and a sphere (right).*

Let us consider a PM of genus 0 (i.e. a plane model topologically equivalent to a sphere). Polygon pasting and vertex joining are able to reduce PM to a primitive PM that has just one vertex and one polygon, but no edges. On the contrary, since polygon cutting is the inverse of polygon pasting and vertex splitting is the inverse of vertex joining, polygon cutting and vertex splitting can be used to build all genus = 0 plane models from a primitive PM, known as skeletal PM which consists of a single vertex and a single face.

We need an operation to create the primitive skeletal PM from nothing. With three operations, skeletal PM creation, polygon cutting, and vertex splitting we can create all genus = 0 models. There is only a single sequence of topologically distinct surface types. Within each surface type, it is possible to use polygon pasting and vertex joining to reduce a PM of a surface to a primitive PM representing that surface.

We use the property that every orientable 2-manifold is topologically equivalent either to the sphere, or to the connected sum of n tori, $n>0$.

To create surfaces other than that equivalent to a sphere we need other operations. These are the so-called global topological operations because they modify more than the local conditions, they modify the topological properties of the surface. In doing so we employ the connected sum and connected minus.

The six abstract operators: polygon cutting, polygon pasting, vertex splitting, vertex joining, connected sum, connected minus form the basis of the so called Euler operators.

The Euler operators act on the data structures representing PM. The six local topological operations, vertex joining and splitting (kev and mev), polygon cutting and pasting (mef and kef) and loop expanding and collapsing (kemr and mekr) can be described by the set of operators and elements listed in Table 1.

Elements	s–solid s–shell f–face r–ring h–hole e–edge v–vertex
Operators	m–make k–kill s–split j - join

Table 1. *Basic elements and operators for the representation of all topologically local and global operations.*

The global operations that modify the model topology could be described according to those operators, the connected sum (kfmrh) and the connected difference (mfkrh).

3.3. Previous work

We can find an in-depth description of the algorithm of marching triangles in recent publications [1,13,14]. These works employ the properties of 3D Delaunay triangulations as well as the 3D Delaunay constraint to get a globally Delaunay triangulation with uniform triangle shape.

We will give a brief description of the original marching triangles method [13]. The algorithm starts with a seed model $M=M_0$ that can be a single triangle. The model M is represented as a list of edges and vertices. The algorithm is implemented as a single pass through the edge list. It does not terminate until all edges have been tested once.

The creation of a new surface vertex x_{proj} to construct a new triangle is performed projecting a constant distance l_{proj} from the selected edge. The projection is perpendicular to the mid-point of the selected edge in the plane of the model boundary element $T_b(x_i, x_k, x_{k+1})$. Next it evaluates x_{new} as the nearest point to x_{proj} on the implicit surface, and applies the Delaunay surface constraint to $T_{new}(x_{k+1}, x_k, x_{new})$. If T_{new} passes the constraint then it is added to the list, otherwise the algorithm tries with adjacent vertices. The Delaunay sphere for the new added triangle is shown in Figure 2.

The algorithm proposed by Hilton and Illingworth fails when the mesh growing starts folding as stated in the work of Akkouche and Galin[1]. The problem is that new created triangles pass the Delaunay test, but some existent triangles can fail this test with this added new point and the added new triangle.

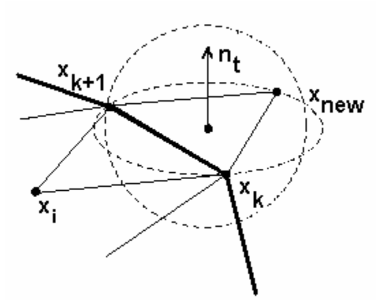


Figure 2. *New triangle creation step.*

Akkouche and Galin have presented an alternative implementation of marching triangles that introduces an adaptive length of the projected distance. Their proposal generalizes the algorithm by adapting the step length to the local curvature of the surface.

4. Our implementation

An implicit surface is represented as the zero-set of points in space that satisfy the equation $f(x) = 0$. The signed distance to the surface is defined by the field function $f(x)$.

We present a tiling method that operates in two stages. The goal of the first one is to obtain a mesh of points that are a good sampling of the final surface. The second stage recover all the surface of the object. The proposed algorithm acts as follows:

1. Surface sampling. Starting with the seed triangle it locates more triangles on the surface, forming a bent surface. This surface is made of glued triangles forming a strip. That is the reason behing the name of the algorithm.

2. Surface filling. It closes the intentionally generated gaps between the strips of triangles.

The algorithm includes two additional stages. One previous stage locates a seed triangle on the implicit surface. In this previous stage we can use an initial model instead of a seed triangle. It could also exist another final stage that subdivides the triangulation using the information of the implicit. The result is a polyhedral surface that approximates the final shape of the object.

4.1. Locating a seed triangle

First of all, an initial seed model is defined, $M = M_0$. This seed model can be a previous model or a single triangle located on the implicit surface. This stage is performed by shifting a triangle through the entire volume. We need one seed triangle for every object in the volume.

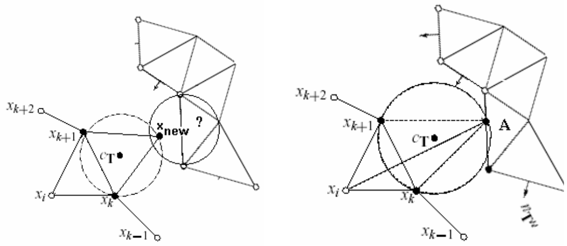
4.2. Surface sampling

In this stage new triangles are added to the initial polyhedral structure constructed with the seed model. This stage was implemented with the 3D polyhedral surface.

Figure 3.a shows the problem of overlapping triangles (see Section 3.3), in this case exists an overlap of the Delaunay spheres. Nevertheless Figure 3.b illustrates a worst situation that implies an overlapping in the triangles which also meet the condition proposed in the initial algorithm of Hilton and Illingworth.

Instead of testing every time all the spheres of the previous triangulated points we propose a new modified criterion that construct the surface by making strips of triangles. The modified criterion includes the Delaunay in-sphere test of the new triangle $T_{new}(x_{k+1}, x_k, x_{new})$ for the region out of the surface.

After projecting a distance l_p in the normal direction to the edge $e(x_{k+1}, x_k)$ in the plane of the triangle $T_b(x_i, x_k, x_{k+1})$ the algorithm leads to a point x_{proj} . Then starting from that point we locate another point on the implicit surface in the normal direction to the triangle $T_{proj}(x_i, x_k, x_{proj})$. The result is a new point x_{new} located on the implicit surface.

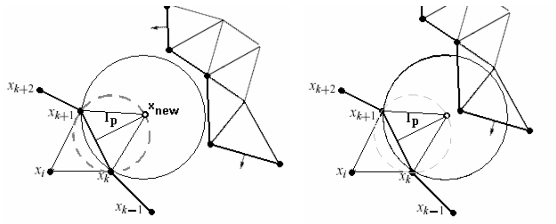


(a) (b)

Figure 3. *Problems detected in the MT algorithm. a) the overlap of the two Delaunay spheres b) overlap of the spheres and the triangles.*

In Figure 4.a we can see the position of the new sphere. We select the point x_{new} as the centre of the sphere. In Figure 4.b we see the case of a near point that meet the Delaunay criterion for all the present triangles. We also exclude this case because it would generate two points very close to each other, and it is not desirable for a good sampling criterion.

The key step of this algorithm is the creation of the new triangle. The solution depends on the marching direction. In Figure 5 we can see the two kinds of striped growth. The fist one that we call spiral growth is shown in Figure 5.a and the second that we call sawline growth is displayed in Figure 5.b.



(a) (b)

Figure 4. *The new modified criterion that overcomes the problem stated by Akkouche and Galin a)test ok, b)test fail.*

This stage have been tested with objects of different topologies. All the objects, the sphere, the pill and the knot have been obtained from a 300x300x300 data size with a projected length of 5 units ($l_p=5$).

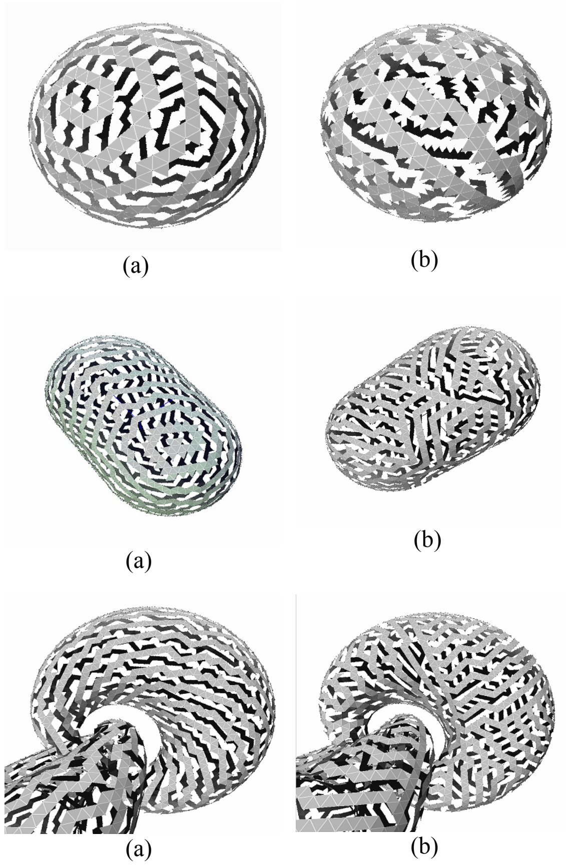


Figure 5. *Two kinds of growth for obtaining a uniform point mesh, it gives a previous approach to the final aspect of the object .a) spiral growth and b) saw line growth. For a sphere (top), pill (centre) and knot surface (bottom).*

In our case the implementation of the spiral growth results in a faster solution due to the number of attempts when a new triangle is created. The results in this case for the sampling are the ones of Table 2. The first row is the size of the implicit data volume used to generate the surface, the second one is the number of points generated by the algorithm, the third row is the number of generated triangles, and the fourth row is the mean distance to the first neighbour obtained by the 3D Delaunay triangulation of the generated points. The last row is the projected distance used to march.

	Sphere	**Pill**	**Knot**
V. Size	300 ^ 3	300 ^ 3	300 ^ 3
Points	5698	1985	5108
Triangles	5696	1983	5106
Mean d.	5.43	6.49	5.46
lp	5	6	5

Table 2. *Results of the sampling with the test objects.*

These results agree with the visual homogeneity of the triangles aspect.. The mean distance is greater than the projected distance due to the algorithm construction of the triangle $T_{new}(x_{k+1}, x_k, x_{new})$ as it could be expected.

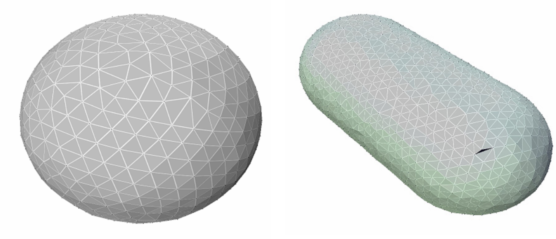
4.3. Surface filling

In this stage we fill the gaps between the strips by testing the consecutive border edges of the polyhedral surface of the previous stage. We only create the triangles that succeed the Delaunay in-sphere test. In this stage the algorithm also advances in strips but only between the existent ones.

The result is shown in Figure 6. The resulting polyhedral surface have two or three border edges that must be closed. In the case of the sphere there are two border edges, and so that we do not see any hole. For the pill there exist three border edges at the end, and that is the reason why we only see one triangular hole in the right hand side of the Figure 6.b. In both cases the result is a correct polyhedral surface.

4.4. Subdivision

We propose a final stage of subdivision to improve the geometry of the final shape. It could be done by using the centre of the triangles to divide the existent ones, locating the new vertex on the position of the isosurface taking into account the normal triangle direction.



(a) (b)

Figure 6. *The result surface after the surface filling stage. a) with the sphere b)with a pill included in a 300x300x300 volume data, with radius 80, and l_p=5.*

5. Results

Visual quality of surfaces is difficult to define and measure because many independent factors such as camera position or shading model can affect the final visualization [17]. In our case we try to develop a fast algorithm with few output triangles. For visualizing the results we have used a software developed in our research group based on the OpenGL libraries. This software provides a smooth surface using the triangle normals, although we have not employed this option to see in detail the constituent triangles of the results.

The marching triangles algorithm reconstructs the correct topology of implicit surfaces features greater than the constant projection distance l_p. Nevertheless the resulting triangulation can be improved with the surface subdivision post-step. We have tested the algorithm and it works properly with data sets of 1 Gigaboxel, being the main problem the volume construction.

5.1. Comparison

The complexity of the implicit surface reconstruction methods can be described as a function of the number of evaluations of the implicit function [10]. The complexity study is similar to the original algorithm [14]. It depends on the size and shape of the object in the data volume. So it also depends on the number of final edges in the polyhedral model $O(N_E)$.

In the first stage, two implicit surface evaluations are needed to find the nearest point located on the implicit surface. In this process we check all the border edges to avoid overlaps at every new triangle creation step.

In the second stage we review all the adjacent border edges to fill the holes between them. The growth is in strips with a hole strip of the same width, the area of the hole is approximately the same as that of the surface tiled, and the evaluations for this stage are $O(N_E)$ as well.

We use a constant projection distance l_p, and hence the element edge and element area are approximately constant. As a result the computational complexity is proportional to the area of the manifold surface reconstructed.

We have selected two algorithms for the comparison with our algorithm. The first one is the adaptive skeleton climbing (ASC) [18], which belongs to the volume based approach. The second one is the marching cubes (MC) [15], which is a usual reference test algorithm.

The results of the reconstruction of the same object with these three algorithms are shown in Figure 7. The size of this data volume is 200x200x200 and all the values are integers. In the case of the *striped* marching triangles (SMT) we can see the effect of discrete values, is improved when the data size is big compared with the projected distance l_p, in this case $l_p=4$. The results on Table 3 have been obtained using an equivalent size for the three algorithms. In the case of MC we have resampled the data volume to a 50x50x50 volume, so the cube cell size is equal to 4.

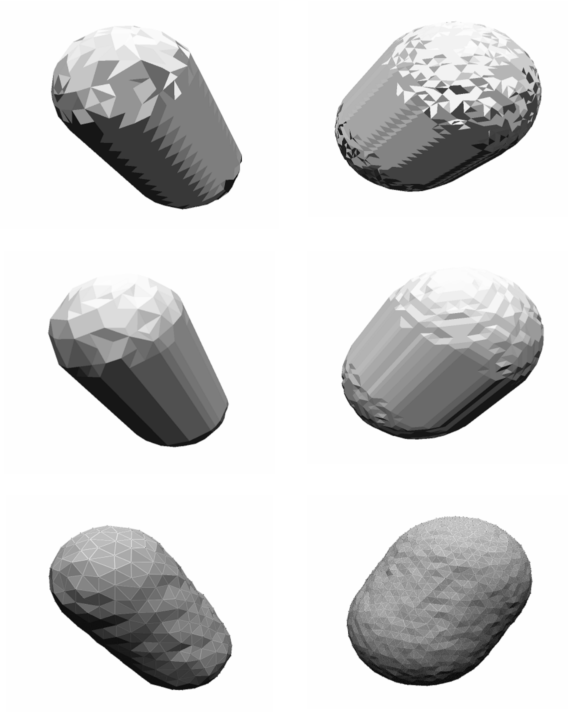


Figure 7. *Comparison of the results according to the aspect to the final triangles. radius 20 (left), and radius 50 (right), in a 200x200x200 volume of integer values. With the algorithms ASC (up), MC(centre), Striped MT (down).*

The used version of ASC has block size *N=4*. For a more detailed comparison between the different ASC block sizes and MC, see the work of Poston et al. [18]. For a more detailed comparison between MC and the original MT algorithm see [13, 14].

As we can observe in Table 3 the number of generated triangles is less in the case of striped marching triangles, but as can be seen in Figure 7, the main problem that shows this algorithm is the low definition in curved areas due to the difficulty of the method with features smaller than the projected distance.

	SMT	ASC	MC
r = 20	1043	1584	1793
r = 30	2025	2984	3417
r = 40	3059	4900	5509
r = 50	4453	7624	8093

Table 3. *Comparisons of the number of triangles depending on the object size, the selected object to this test is the pill with radius r.*

6. Conclusions and future work

Surface reconstruction using Marching Triangles offers a possible solution to the interactive modelling. In this and other scopes the speed of the algorithm is very important.

In this work we have proposed a modification of the marching triangles algorithm of Hilton and Illingworth witch overcomes the existent problem of the overlapping triangles.

There has been reached a good sampling of the surface in a first stage of our algorithm. In this first stage we have also obtained the triangulation of the half surface. Leaving the other half surface for the second stage. This triangles can be used as a preview of the final shape of the object.

A new modified criterion for the triangle creation has been proposed that leads to a valid surface sampling for any object topology.

The Euler operators have been used to extend the surface, so the constructed surface is a correct polyhedral surface.

Our current research is oriented to finding the correspondence between the plane models and the reconstructed surfaces in order to know when it will be needed a new stage of crack closing. It would be also very interesting to study the possibility of using the correspondence information for texture mapping. We are also working in an adaptive version using a curvature dependent process of subdivision which solves the problem illustrated in Figure 7.

Acknowledgements

The authors would like to thank their colleagues in the Artificial Vision Group, and special thanks to V. Brea and M.J. Gonzalez for useful suggestions and their advice on the paper. This research was supported by CICYT TIC 200-0399-C02-02 and Xunta de Galicia PGIDT01TIC20601PN grants.

References

1. S. Akkouche, E. Galin. "Adaptive Implicit Surface Polygonization Using Marching Triangles". In *Computer Graphics Forum*, v. 20(2), pp. 67-80, 2001.
2. A. Allgower and K. Georg. "Numerical Continuation Methods", Springer-Verlag, New York, 1990.
3. D. Baraff. "Linear-time dynamics using Lagrange multipliers". In SIGGRAPH 96 Conference Proceedings, *Computer Graphics Proceedings*, Annual Conference Series, pp. 137-146. ACM SIGGRAPH, Addison Wesley, August 1996.
4. D. Baraff and A. Witkin. "Large steps in cloth simulation". *In SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pp. 43-54. ACM SIGGRAPH, July 1998.
5. J. Bloomenthal, "Polygonization of Implicit Surfaces", *Computer Aided Geometric Design*, v. 5, pp. 341–355, 1988.
6. J.-D. Boissonnat. "Geometric structures for three dimensional shape representation". *ACM Transactions on Graphics*, v. 3, 266-286, October 1984.
7. A. Bottino, W. Nuij, and van Overveld, K. "How to shrinkwrap through a critical point: an algorithm for the adaptive triangulation of iso-surfaces with arbitrary topology". In *Proc. Implicit Surfaces* '96, pp. 53-72, Oct. 1996.
8. Carnegie Mellon University. Lectures-Handouts. "The Euler Operators" 2002. http://weld.arc.cmu.edu/48-745/Index4.html
9. CGAL. Computacional Geometry Algorithms Library. "Basic Library". Release 2.4, May. 2002. http://www.cgal.org
10. M. Cani "Implicit Representations in Computer Animation : a Compared Study", *Proceedings of Implicit Surfaces* '99, Sep 1999. http://www-imagis.imag.fr/Publications/1999/Can99
11. G. Debunne, M. Desbrun, A. Barr, and M. Cani. "Interactive multiresolution animation of deformable models". In N.Magnenat-Thalmann and D. Thalmann, editors, *Eurographics Workshop on Computer Animation and Simulation* '99, pp. 133-144. Springer-Verlag, Sept. 1999.
12. M.P. Gascuel and M. Desbrun. "Animation of Deformable models using implicit surfaces", IEEE *Transactions on Vision and Computer Graphics*. V.3(1) pp. 39-50, March.1997.
13. A. Hilton, A. Stoddart, J. Illingworth, and T. Windeatt, "Marching triangles: Range image fusion for complex object modelling", in *Proc. of IEEE International Conference on Image Processing*, Lausane, v. 2, pp. 381-384, 1996.
14. A. Hilton, J. Illingworth, "Marching triangles: Delaunay Implicit Surface Triangulation" Note: Technical Report for CVSSP. http://www.ee.surrey.ac.uk/Research/VSSP/3DVision/hilton-mt-97.ps.gz
15. W. E. Lorensen, H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm", *Computer Graphics*, V.21(4), pp. 163-169, 1987.
16. M.E. Mortenson. "Geometric Transformations". Industrial Press, December 1995.
17. P. Ning and J. Bloomenthal, "An evaluation of implicit surface tilers", *IEEE Computer Graphics and Applications*, v.13(6), pp. 33-41, 1993.
18. T. Poston, Tien-Tsin Wong, and Pheng-Ann Heng. "Multiresolution isosurface extraction with adaptive sceleton climbing". *EUROGRAPHICS*, 17(3), 1998.
19. B. T. Sander and J. C. Hart, "Guaranteeing the topology of an implicit surface polygonization for interactive modeling." *Computer Graphics Proceedings*, ACM/SIGGRAPH, pp. 279-286, 1997.
20. B. T. Sander "Polygonizing Implicit Surfaces with Guaranteed Topology". PhD thesis, School of EECS, Washington State University, May 1997.
21. F. Triquet, P. Meseure, C. Chaillou. "Fast Polygonization of Implicit Surfaces" *WSCG'2001*, v. 2, pp. 283-290, Feb 2001.
22. A. P. Witkin and P. S. Heckbert. "Using particles to sample and control implicit surfaces". Computer Graphics. *Proceedings of SIGGRAPH 94*, v. 12, pp. 269-278, July 1994.
23. R. Westermann, L. Kobbelt, and T. Ertl. "Real-Time Exploration of Regular Volume Data by Adaptive Reconstruction of Isosurfaces". *The Visual Computer*, v. 15 pp. 100-111, 1999.