Cermak,M., Skala,V.: Accelerated Edge Spinning Algorithm for Implicit Surfaces, ICCVG2002 Int.Conf., Poland, ISBN 839176830-9, pp.174-179, 2002

INTERNATIONAL CONFERENCE ON

COMPUTER VISION AND GRAPHICS

With Special Session for Young Scientists
and
Summer School on Image Processing

Zakopane, Poland, September 25-29, 2002

Conference Proceedings
Volume One

Organizers:

Association for Image Processing
Poland

Silesian University of Technology
Gliwice, Poland

Institute of Theoretical and Applied Informatics,
PAS, Gliwice, Poland

Programme committee:

S. Ablameyko (BY)
P. Bhattacharya (US)
E. Bengtsson (SE)
A. Borkowski (PL)
D. Chetverikov (HU)
L. Chmielewski (PL)
J. Chojcan (PL)
R. Choras (PL)
S. Dellepiane (IT)
M. Domanski (PL)
U. Eckhardt (DE)
A. Gagalowicz (FR)
E. Grabska (PL)
H. Heijmans (NL)
J.M. Jolion (FR)
A. Kasinski (PL)
R. Klette (NZ)
W. Kosinski (PL)
R. Kozera (AUS)
H. Kreowski (DE)
Z. Kulpa (PL)
M. Kurzynski (PL)
W. Kwiatkowski (PL)
G. Levina (RU)
L. Luchowski (PL)
R Lukac (SK)
V. Lukin (UA)
W. Malina (PL)
A. Materka (PL)
H. Niemann (DE)
M. Nieniewski (PL)
L. Noakes (AUS)
M. Orkisz (FR)
M. Paprzycki (US)
J. Piecha (PL)
K. Plataniotis (CND)
H. Palus (PL)
D. Paulus (DE)
J. Roerdink (NL)
P. Rokita (PL)
R. Sara (CZ)
V. Skala (CZ)
B. Smolka (PL)
J. Soldek (PL)
G. Stanke (DE)
R. Tadeusiewicz (PL)
V. Valev (BG)
T. Vintsiuk (UA)
J. Zabrodzki (PL)
M. Zaremba (CND)

Local organizing committee:

A. Bal (PL), D. Bereska (PL), P. Lagodzinski (PL), A. Matulewicz (PL), H. Palus (PL), B. Smolka (PL) M. Szczepanski (PL), G. Zajączkowski (PL)

Invited lectures:

A. Gagalowicz
P. Kiciak
R. Kozera
K. Myszkowski
M. Orkisz,
A. Śluzek

Editor

Konrad Wojciechowski
President of Association for Image Processing

Cermak,M., Skala,V.: Accelerated Edge Spinning Algorithm for Implicit Surfaces, ICCVG2002 Int.Conf., Poland, ISBN 839176830-9, pp.174-179, 2002

Martin CERMAK[1], **Vaclav SKALA**
University of West Bohemia, Department of Computer Science and Engineering, Faculty of Applied Sciences, Univerzitni 8, 306 14 Plzen
{cermakm|skala}@kiv.zcu.cz

ACCELERATED EDGE SPINNING ALGORITHM FOR IMPLICIT SURFACES

Abstract.

This paper presents a new fast modification of one method for polygonization of implicit surfaces. The algorithm puts emphasis on the quality of polygonal mesh and on the polygonization speed. The implementation is not complicated and only the standard data structures are used. This algorithm is based on the surface tracking scheme and it is compared with the Marching cubes algorithm that is based on the similar principle. Experimental results prove quality and efficiency of acceleration technique used for such type of geometric algorithms.

keywords : *implicit surfaces, polygonization, triangulation, acceleration, edge spinning, marching method.*

1 INTRODUCTION

Implicit surfaces become widely used in several applications in computer graphics and visualization. An implicit surface is mathematically defined as a set of points in space **x** that satisfy the equation $f(\mathbf{x})=0$. Thus, visualizing implicit surfaces typically consists in finding the zero-set of f, which may be performed either by polygonizing the surface or by direct ray-tracing.

There are two different definitions for implicit surfaces. The first one [2], defines an implicit object as $f(\mathbf{x})<0$ and the second one, F-rep [7], [10], defines it as $f(\mathbf{x})\geq 0$. In our implementation, we use the F-rep definition of implicit objects.

Existing polygonization techniques may be classified into three categories. Spatial sampling techniques that regularly or adaptively sample the space to find the cells that straddle the implicit surface [2], [8]. Surface tracking approaches (also known as continuation methods) iteratively create a triangulation from a seed element by marching along the surface

[1] This work was supported by the Ministry of Education of the Czech Republic - project MSM 235200005

Cermak,M., Skala,V.: Accelerated Edge Spinning Algorithm for Implicit Surfaces, ICCVG2002 Int.Conf., Poland, ISBN 839176830-9, pp.174-179, 2002

[1], [2], [5], [6]. Surface fitting techniques progressively adapt and deform an initial mesh to converge to the implicit surface.

2 ACCELERATED EDGE SPINNING ALGORITHM

The Edge Spinning algorithm [3] is a new method for polygonization of implicit surfaces. The main advantage is that the algorithm generates a well-shaped and correct triangular mesh. The accurate coordinates of new generated points are determined by spinning of edges of already generated boundary triangles. The polygonization speed is less then the well-known Marching Cubes algorithm, see Fig. 5, which is simple, fast but it generates badly-shaped triangles. Therefore, the modification for speed-up is obvious.

2.1 Data structures

The presented algorithm uses only the standard data structures used in computer graphics. The main data structure is an edge that is used as a basic building block for polygonization. We use the standard winding edge and therefore, the resulting polygonal mesh is correct and complete with neighborhood among all triangles generated. If a triangle's edge lies on the triangulation border, it is contained in the *list of active edges* (dynamically allocated list) and it is called as an *active edge*. Each point, which is contained in an active edge, contains two pointers to its left and right active edge (left and right directions are in active edges' orientation) and the index of sub-area in which the point lies as well. Each sub-area has its own dynamically allocated list of points which lie in. Sub-areas are implemented as an array and each of them has its unique index.

2.2 The basic idea of the algorithm

The Edge spinning algorithm consists of several steps bellow. Detailed description of the original algorithm is introduced in [3]. In this situation, it is necessary to explain principally step 6 which is important for understanding the problem. Step 6 is the most time consuming part of the algorithm, see Fig. 5, and our modification of the algorithm is aimed at it.

1. Find a starting point $\mathbf{p}_0$.
2. Create a first triangle T_0, see Fig. 1a.
3. Include the edges (e_0, e_1, e_2,) of the first triangle T_0 into the active edges list.
4. Polygonize the first active edge *e* from the active edges list.
5. Delete the actual active edge *e* from the active edges list and include the new generated active edges at the end of the active edges list.
6. Check the distance between the new generated point $\mathbf{p}_{new}$ and all the other points which lie on the border of already triangulated area (which lie in all the other active edges).
7. If the active edges list is not empty return to step 4.

Cermak, M.; Skala, V.: Accelerated Edge Spinning Algorithm for Implicit Surfaces, ICCVG2002 Int.Conf., Poland, ISBN 839176830-9, pp.174-179,

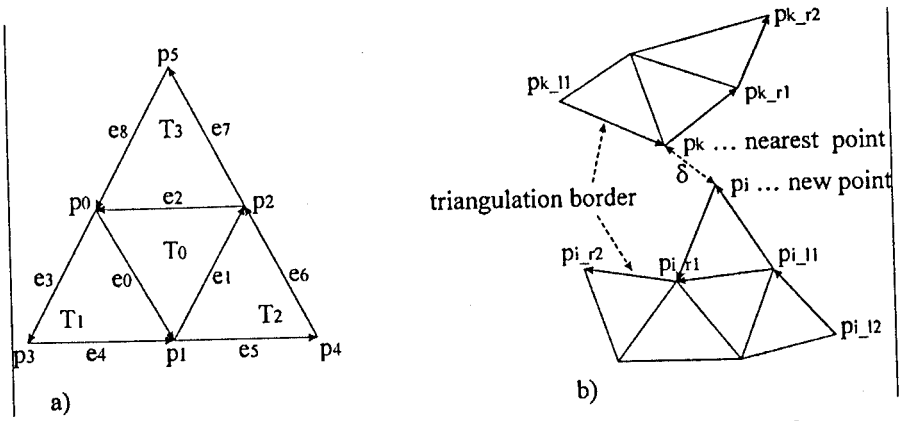

Fig. 1 a) The first steps of the Marching triangles algorithm; b) Distance δ between the new point p_i and the nearest point p_k.

2.3 Distance test

The Edge spinning algorithm marches over the surface of an implicit object. This principle is known as a continuation scheme. During polygonization, the new included point into the triangular mesh can lie near to the triangulation border (border of already triangulated area), see Fig. 1b. The algorithm looks for the nearest point $\mathbf{p_k}$, to the new point $\mathbf{p_i}$, which lies on the triangulation border. If the distance δ is less then δ_{lim} then the algorithm has to solve this situation, detailed description is in [3]. The original algorithm checks distances with the algorithm complexity *O(N)*, where *N* is a number of points on the triangulation border.

2.4 Space subdivision principle

The original distance check algorithm takes more time if required scene details grow (growing number of points on the triangulation border). In case that the new included point can lie near to any point of the boundary, it is not possible to determine some subset of candidates to the nearest point ahead.

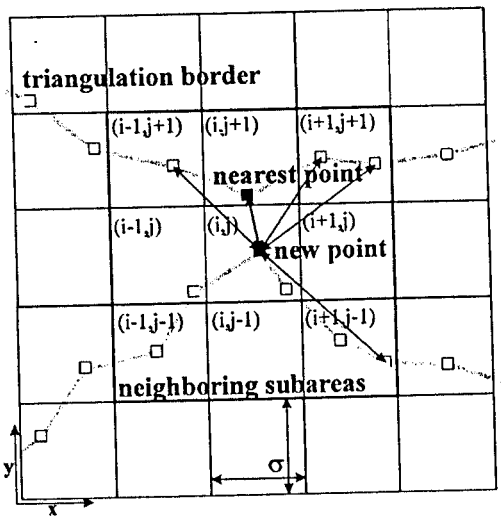

Fig. 2 The space subdivision scheme for distance checking.

Cermak,M., Skala,V.: Accelerated Edge Spinning Algorithm for Implicit Surfaces, ICCVG2002 Int.Conf., Poland, ISBN 839176830-9, pp.174-179, 2002

Advantageous solution is dividing of space into sub-spaces (sub-areas). Then, the nearest point must lie in the same sub-area like the new included point or in the closest neighborhood. In order to validity of this theorem the next equation must be valid as well: $\sigma \geq \delta_{lim}$, where σ is the size of sub-areas (cube shape), see Fig. 2, and δ_{lim} is the limit distance for distance check.

The algorithm complexity of this solution is *O(M)*, where *M* is a number of points in adjacent sub-areas and *M<<N*. Fig. 2 shows 9 possible sub-areas in E^2 case, there are 27 possible sub-areas in E^3 case.

3 EXPERIMENTAL RESULTS

All the experiments were accomplished on the implicit object *Genus*, see Fig. 3. Its implicit function is described as follows:

$$f(\mathbf{x}) = r_z^4 \cdot z^2 - \left[1 - (x/r_x)^2 - (y/r_y)^2\right] \cdot \left[(x - x_1)^2 + y^2 - r_1^2\right] \cdot \left[(x + x_1)^2 + y^2 - r_1^2\right] = 0, \qquad (1)$$

where $\mathbf{x}=[x, y, z]$ and the parameters are: $r_x=6$, $r_y=3.5$, $r_z=4$, $r_1=1.2$, $x_1=3.9$.

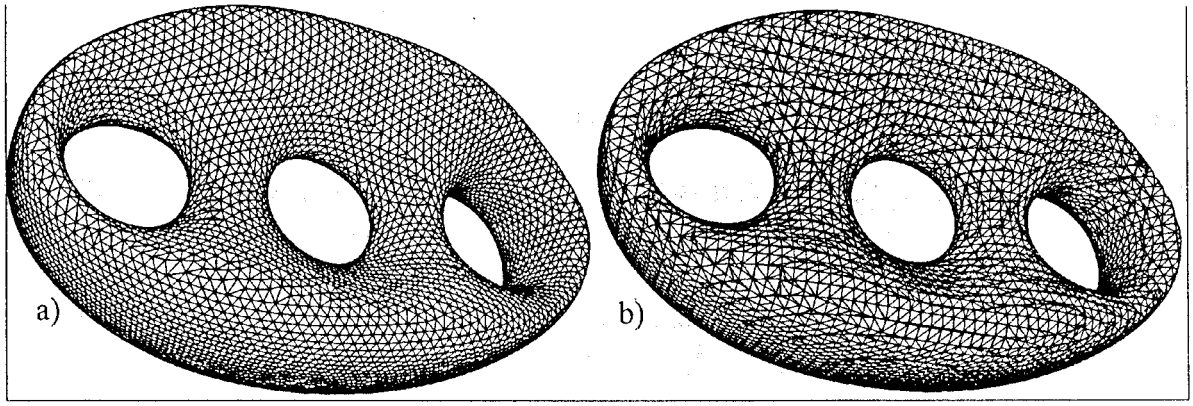

Fig. 3 The triangular mesh of the implicit object Genus that is polygonized by a) the Edge spinning algorithm, b) the Marching cubes algorithm.

A visual comparison of a triangle's shape quality proves that the polygonal mesh generated by the Edge spinning algorithm is much better than the Marching cubes method. This result is vindicated by the next histogram, Fig. 4, which shows the percentage frequency of triangles' angles generated in the output polygonal mesh. The histogram demonstrates that the acceleration technique used has not negative effect on the triangles' shape.

The original Edge spinning (ES) algorithm takes more time for polygonization of implicit objects than the Marching cubes (MC) algorithm. This deficiency is successfully remedied in the described modified version. Fig. 5 shows the polygonization time's ratios between mentioned algorithms. This diagram proves that the original Edge spinning algorithm is noticeably slower than the Marching cubes method but the accelerated Edge spinning (ESA) algorithm is stably (constantly) faster than the Marching cubes algorithm in all experiments. It demonstrates the efficiency of the acceleration technique used.

Cermak,M., Skala,V.: Accelerated Edge Spinning Algorithm for Implicit Surfaces, ICCVG2002 Int.Conf., Poland, ISBN 839176830-9, pp.174-179,

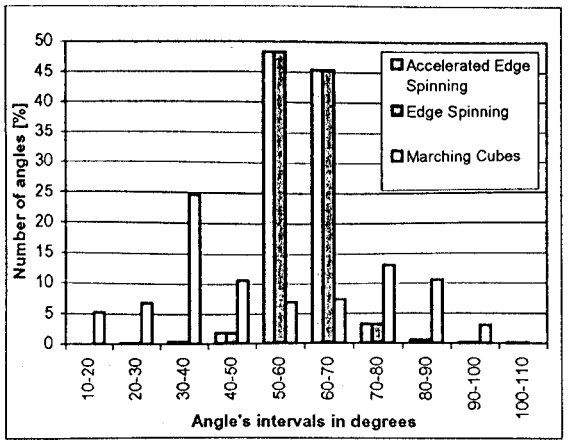

Fig. 4 Histogram of the triangle shape quality for the ES, the accelerated ES and the MC algorithms. Generated for N=1000, see Tab. 1.

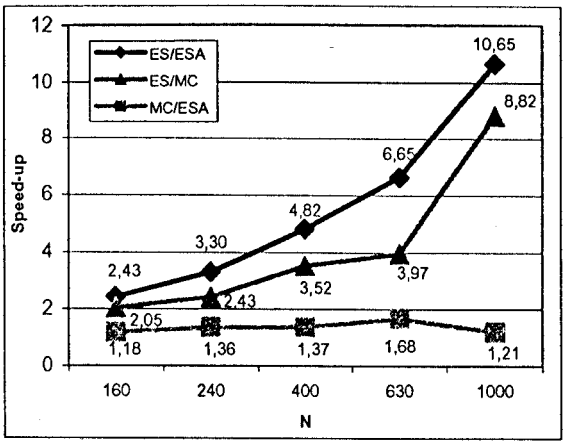

Fig. 5 Computational time comparison (speed-up) between the original ES algorithm and the accelerated one, between the original ES and the MC algorithm and between the ESA and the MC method.

The computational time of the Edge spinning algorithms consists of two main parts, the *edge polygonization time* and the *distance test time*. Fig. 6 shows the ratio between both parts for the original Edge spinning algorithm and the accelerated one as well.

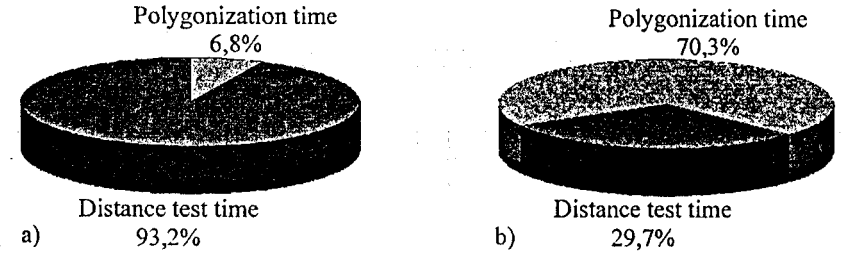

Fig. 6 The time ratio between the Polygonization time and the Distance test time; a) the original Edge spinning algorithm, b) the accelerated Edge spinning algorithm.

It is obvious that the algorithm complexity of the distance checking is significantly decreased and the acceleration technique used is effective for such type of geometric problems.

Tab. 1 The measured values from experiments.

	N	160	240	400	630	1000
Edge spinning	Triangles:	53 992	120 766	335 622	834 966	2 110 538
	Vertices:	26 992	60 379	167 807	417 479	1 055 265
	Time [ms]:	922	2 874	12 848	49 912	225 294
Marching cubes	Triangles:	69 676	157 520	437 800	1 085 376	2 735 836
	Vertices:	34 826	78 756	218 896	542 684	1 367 914
	Time [ms]:	450	1 182	3 646	12 588	25 546

Tab. 1 contains the measured values from all experiments described above. The values indicate that the Edge spinning algorithm generates about 23% triangles less than the

Cermak,M., Skala,V.: Accelerated Edge Spinning Algorithm for Implicit Surfaces, ICCVG2002 Int.Conf., Poland, ISBN 839176830-9, pp.174-179, 2002

Marching cubes method and the output polygonal mesh consists of well-shaped triangles. The presented results are original and were verified on many nontrivial implicit surfaces as well.

The N variable, used in figures above and in Tab. 1, represents a desired object detail. Specifically, the average triangles edges' length is proportional to N and to computing area's size as well, i.e. with growing N the triangles' edges are shorter and more of them can be held in line in computing area. The computing area's size used in experiments is $[<x_{min},x_{max}>,<y_{min},y_{max}>,<z_{min},z_{max}>]=[<-8,8>,<-8,8>,<-8,8>]$.

4 CONCLUSIONS

In this paper, we have presented the new fast modification of the Edge spinning algorithm for polygonization of implicit surfaces. Experimental results proved that the selection of subset of candidate points by the space subdivision scheme is an effective way for this type of geometric algorithms. The Edge spinning algorithm generates a well-shaped triangular mesh and the polygonization speed is comparable with well-known and simple Marching cubes algorithm now. The presented new algorithm was tested on several objects and proved stability and properties described above. In next research, we will work on adapting of the Edge spinning algorithm to local curvature of an implicit surface.

REFERENCES

[1] Akkouche, S., Galin, E.: *Adaptive Implicit Surface Polygonization using Marching Triangles*, Computer Graphic Forum, 20(2): 67-80, 2001.

[2] Bloomenthal, J.: *Graphics Gems IV*, Academic Press, 1994.

[3] Cermak, M., Skala, V.: *Polygonization by the Edge Spinning*, accepted for publication, Algoritmy 2002. http://herakles.zcu.cz/~cermakm/papers/cermak_alg2002.pdf

[4] Cermak, M., Skala, V.: *Space Subdivision for fast Polygonization of Implicit Surfaces*, accepted for publication, Herlany, ECI 2002. http://herakles.zcu.cz/~cermakm/papers/cermak_eci2002.pdf

[5] Hartmann, E.: *A marching method for the triangulation of surfaces*, The Visual Computer (14), pp.95-108, 1998.

[6] Hilton, A., Stoddart, A.J., Illingworth, J., Windeatt, T.: *Marching Triangles: Range Image Fusion for Complex Object Modelling*, Int. Conf. on Image Processing, 1996.

[7] "Hyperfun: Language for F-Rep Geometric Modeling", http://cis.k.hosei.ac.jp/~F-rep/

[8] Ohraje, Y., Belyaev, A., Pasko, A.: *Dynamic meshes for accurate polygonization of implicit surfaces with sharp features*, Shape Modeling International 2001, IEEE, 74-81.

[9] Rousal, M., Skala, V.: *Modular Visualization Environment - MVE*, Int. Conf. ECI 2000, Herlany, Slovakia, pp.245-250, ISBN 80-88922-25-9.

[10] Rvachov, A.M.: *Definition of R-functions*, http://www.mit.edu/~maratr/rvachev/p1.htm