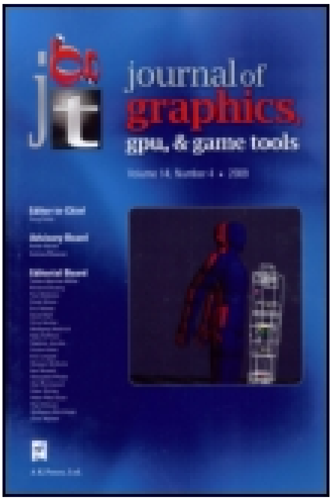

Journal of Graphics Tools
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ujgt19

Triangulating Convex Polygons Having T-Vertices

P. Cignoni [a], C. Montani [b] & R. Scopigno [c]
[a] IEI-CNR, Via S. Maria 46, 56126, Pisa, Italy E-mail:
[b] IEI-CNR, Via S. Maria 46, 56126, Pisa, Italy E-mail:
[c] CNUCE-CNR, Via S. Maria 36, 56126, Pisa, Italy E-mail:
Published online: 06 Apr 2012.

Triangulating Convex Polygons Having T-Vertices

P. Cignoni
IEI-CNR

C. Montani
IEI-CNR

R. Scopigno
CNUCE-CNR

Abstract. A technique to triangulate planar convex polygons having T-vertices is described. Simple strip or fan tessellation of a polygon with T-vertices can result in zero-area triangles and compromise the rendering process. Our technique splits such a polygon into one triangle strip and, at most, one triangle fan. The technique is particularly useful in multiresolution or adaptive representation of polygonal surfaces and the simplification of surfaces.

Triangulating a planar convex polygon is a simple but important task. Although modern graphics languages support the planar convex polygon as an elementary geometric primitive, better performances are obtained by breaking down polygons into compact sequences of triangles. The triangle-strip (Figure 1(a)) and triangle-fan methods (Figure 1(b)) are compact and efficient ways to represent the sequences of triangles and are provided as output primitives in most graphics libraries (OpenGL [Neider et al. 93] and, partially, PHIGS [Gaskins 92], for example).

1086-7651/96 $0.50 per page

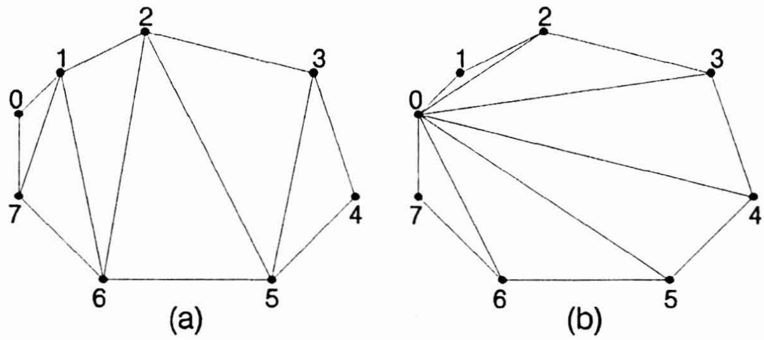

Figure 1. Triangulating a convex polygon: (a) a triangle strip with vertices $v_0, v_7, v_1, v_6, v_2, v_5, v_3, v_4$ and (b) a triangle fan with vertices $v_0, v_7, v_6, v_5, v_4, v_3, v_2, v_1$.

The triangulation becomes a little more complicated when one or more vertices of the polygon are T-vertices, i.e. three or more aligned vertices exist. This problem arises, for example, in the adaptive representation of polygonal surfaces ([Muller, Stark 93], [Haemer, Zyda 91]); one of the simplest algorithms to avoid cracks (small holes between polygons at different levels of resolution) is to move some of the vertices of the lower resolution polygons onto the boundary of the higher ones, thus introducing T-vertices. In these cases, naively creating a triangle strip or fan, as shown in Figure 1, can yield degenerate (zero-area) triangles.

In this article, we present a simple technique which allows us to split convex polygons with T-vertices into one triangle strip and, possibly, one triangle fan. We assume that the user is acquainted with the existance of T-vertices and with their position on the polygon's boundary (this is not a limitation in practical cases), and that the starting vertex of the chain is not a T-vertex (a simple circular shift of the vertices can solve this problem).

The algorithm is shown in Figure 2. Here, the indices *front* and *back* assume the values $0, 1, \cdots$, and $n-1, n-2, \cdots$, respectively. Starting from a *regular* vertex (non-T-vertex) of the polygon, the algorithm adds vertices to the triangle strip in the usual zigzag way (Figure 2(a)) until just one unprocessed regular vertex remains. Then, three different cases can occur:

1. The regular vertex is the only vertex to be processed (Figure 2(b)). It can be added as the last point of the strip. There is no need for a triangle fan.

2. The vertex candidate for the insertion into the strip is the regular one (Figure 2(c)). A triangle fan is built. The sequence of vertices of the fan assures the correct triangles' orientation.

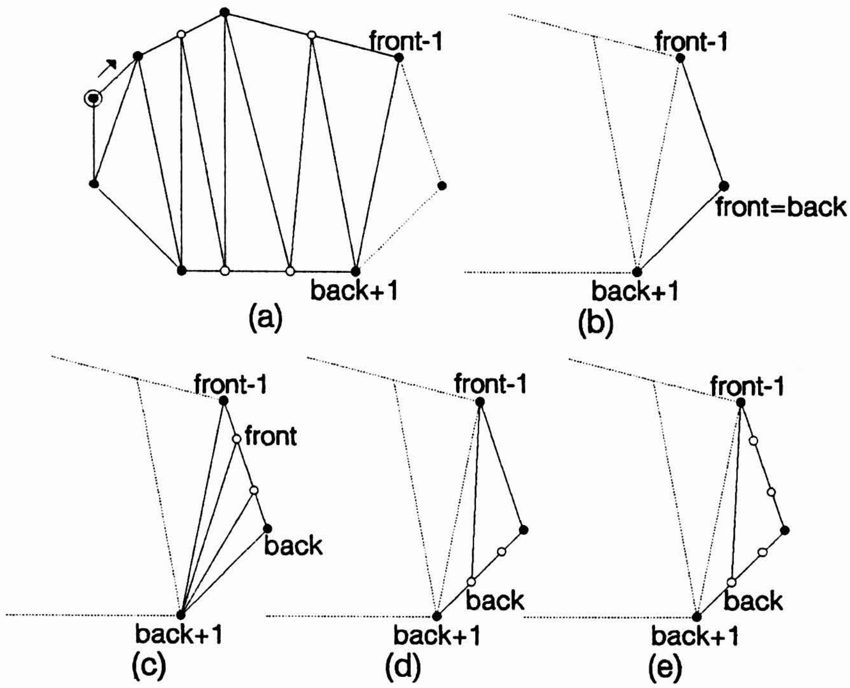

Figure 2. Triangulating a convex polygon with T-vertices (the vertices with a white circle). (a) The construction of the strip interrupts when just one unprocessed regular vertex (the black vertex on the right) is left; (b) if the regular vertex ($v_{front=back}$) is the only vertex to be processed, then it is added as last point of the strip; (c) the next vertex to be processed (v_{back}) is the regular one: a triangle fan is built (in the example the fan $v_{back+1}, v_{back}, \cdots, v_{front}, v_{front-1}$); (d), (e) the candidate vertex (v_{back}) is a T-vertex: it is added to the strip and the algorithm continues until either (b) or (c) occurs.

3. The vertex candidate for the insertion into the strip is a T-vertex (Figure 2(d) or 2 (e)). The construction of the triangle strip continues until one of the previous situations occurs.

Our simple triangulation is computed in linear time. A Delaunay triangulation of a convex polygon can also be computed in linear time [Aggarwal et al. 89]; our technique is not optimal with respect to the *best* shape of the possible resulting triangles. However, we think the simplicity and effectiveness of our solution makes it valuable in practical cases. The algorithm has been used extensively in the implementation of the DiscMC [Montani et al. 95] [Montani et al. 94], a public domain software for the extraction, simplification, and rendering of isosurfaces from high resolution regular data sets.

References

[Aggarwal et al. 89] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor "A Linear Time Algorithm for Computing the Voronoi Diagram of a Convex Polygon." *Discrete and Computational Geometry*, 4:591–604(1989).

[Gaskins 92] T. Gaskins *PHIGS Programming Manual.* Sebastopol, CA: O'Reilly & Associates, 1992.

[Haemer, Zyda 91] M. D. Haemer and M. Zyda "Simplification of Objects Rendered by Polygonal Approximations." *Computers & Graphics*, 15(2):175–184(1991).

[Montani et al. 94] C. Montani, R. Scateni, and R. Scopigno "Discretized Marching Cubes." In *Visualization '94 Proceedings* edited by R. Bergeron and A. Kaufman, pp. 281–287. Washington, DC: IEEE Computer Society Press, 1994.

[Montani et al. 95] C. Montani, R. Scateni, and R. Scopigno. "Decreasing Isosurface Complexity via Discretized Fitting." *Tech. Rep. B4-37-11-95.* Pisa, Italy: IEI-CNR, November 1995.

[Muller, Stark 93] H. Muller and M. Stark "Adaptive Generation of Surfaces in Volume Data." *The Visual Computer*, 9(4):182–199(1994).

[Neider et al. 93] J. Neider, T. Davis, and M. Woo. *OpenGL Programming Guide.* Reading, MA: Addison Wesley, 1993.

Web Information: http://miles.cnuce.cnr.it/cg/homepage.html
DiscMC is available on http://miles.cnuce.cnr.it/cg/swOnTheWeb.html

P. Cignoni, IEI-CNR, Via S. Maria 46, 56126 Pisa, Italy (cignoni@iei.pi.cnr.it)

C. Montani, IEI-CNR, Via S. Maria 46, 56126 Pisa, Italy (montani@iei.pi.cnr.it)

R. Scopigno, CNUCE-CNR, Via S. Maria 36, 56126 Pisa, Italy (r.scopigno@cnuce.cnr.it)

Received March 11, 1996; accepted May 6, 1996