On the Quality of Point Set Triangulations based on Convex Hulls

Peter Jenke, Anders Hast, Stefan Seipel

Abstract

In this paper we describe a method for directly generating triangle strips from unstructured point clouds based on onion peeling triangulation (OPT). It is an iterative reconstruction of the convex hulls of point clouds in the 2D plane, and it uses pairs of subsequent layers to establish triangle strips. We compare the obtained triangulations with the results of Delaunay triangulations in terms of the distribution of the symmetry of obtained triangles and in regard to the number of polygons/vertices emitted. Our initial results show that onion peeling is a straightforward method to directly obtain large triangle strips of point clouds. As expected, the triangulation is not as well behaved as in Delaunay-triangulation [VK07]. In terms of triangle complexity and average strip length OPT is a very favorable triangulation alternative which also lends suitable for the triangulation of 3D point clouds.

Keywords: Surface reconstruction, triangle strip, convex layers, rotating calipers

1. Introduction

The triangulation of large unstructured point clouds is a relevant issue in many applications that use e.g. range imaging data. The result of any such point triangulation should be a well behaved tessellation of the object surface, and ideally it should minimize vertex data flow in the rendering pipeline. This is usually accomplished by constructing triangle strips [AHMS94]. Several methods for constructing triangle strips from a point cloud are known: Some of them operate directly on the point set; others work indirectly, by first computing a triangulation of the point set (usually a mesh) and then applying a stripification algorithm on the result. A method which creates triangle strips from a mesh is presented by Gopi and Eppstein [GE04]: This algorithm searches for a perfect matching in the dual graph of the given triangulation. By the matching, every triangle in the mesh is paired with exactly one of the adjacent triangles. From these pairs, a cycle of connected triangles through the mesh is constructed; the result is a triangle strip. Boubekeur et al. propose a method called surfel strips [BRS05], which aims at producing a number of small triangle strips. This method processes the point set directly, whereby it is partitioned into a number of subsets, using an Octree. For a subset of points, a closest distance plane is fitted upon which points are projected. The actual triangulation is performed by Delaunay triangulation, followed by subsequent stripification.

Arkin presents a method, which operates directly on the point set [AHMS94]. An initial triangulation is created by connecting some interior point with points on the convex hull. Any non-empty triangle of this triangulation is then subdivided by choosing a point inside this triangle and connecting it with the vertices of the current triangle. This process is repeated until all triangles are empty.

2. Triangle strips from Convex Hulls

The convex layers of a planar point set [Cha85] are constructed by incrementally computing the convex hull of the set, calculating the set difference between the set and the hull and computing the hull for the remaining points. This process (also known as onion peeling) is repeated until the set difference is empty.

For creating triangle strips from a set of convex layers, a method based on the rotating calipers [Tou83] is used: Given a convex polygon P. Two parallel lines of support l_1, l_2 through two points p_1, p_2 of P are rotated until both touch another point: p_1 touches p'_1 and p_2 touches p'_2 (figure 1). Then the angles by which the lines were rotated are compared: The smaller one determines the new direction of the calipers. The lines are moved to the next point pair and the process is repeated until p_1, p_2 are reached again. The algorithm for triangulating an annulus of a set of convex layers works as follows [Tou86]: An annulus has an outer hull A and an inner hull B. The algorithm creates a sequence of

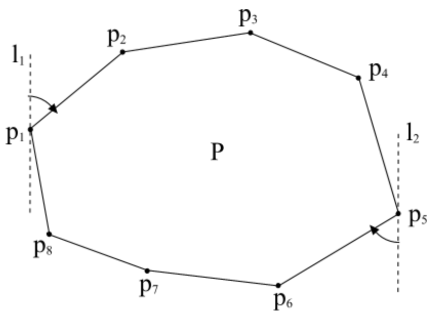

Figure 1: *Rotating calipers.*

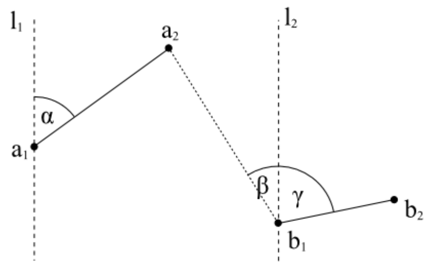

Figure 3: *Triangulating annuli:* $\beta \leq 0$ *and* $\alpha < \gamma$.

points, describing the edges of the triangulation where one point is member of A and the other belongs to B. This set of edges is denoted as R.

As a first step, the leftmost points in A and in B – a_1 and b_1 – are found. Two vertical lines of support are defined: l_1 goes through a_1, l_2 through b_1. Three angles are together the criterion, to determine between which points the next edge of the triangulation is placed (figure 2):

- α - enclosed by the line a_1 - a_2 and l_1
- β - enclosed by the line b_1 - a_2 and l_2
- γ - enclosed by the line b_1 - b_2 and l_2

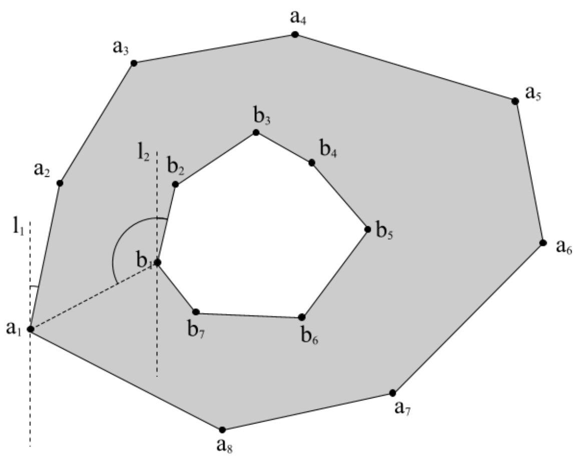

Figure 2: *Triangulating annuli: Startconfiguration.*

The next edge of the triangulation is chosen depending on the angles α, β and γ: If $\beta \leq 0$, the point which can be 'hit' from a_1 is a_2 and from $b_1 - b_2$. The new edge depends on the first touched point: If $\alpha < \gamma$, l_1 hits a_2 before l_2 hits b_2 and the next edge is $a_2 - b_1$ (figure 3). In this case, the rotation center in A is moved to a_2, i.e. l_1 is in the next step rotated around a_2. The line l_2 is still rotated around b_1. The slope of both lines is now the same as the slope of line $a_1 - a_2$. If $\alpha > \gamma$, b_2 is hit first by l_1. Now, the triangulation takes the opposite direction – the new edge is $a_1 - b_2$ and the center of rotation for l_2 moves from b_1 to b_2. In the third case is $\alpha = \gamma$; both a_2 and b_2 are hit at the same time by l_1 and l_2, respectively. Two new edges are added ($a_2 - b_2$ and $b_1 - a_2$) and the center of rotation is moved for both lines.

If $0 < \beta$, the triangulation depends on the angles β and γ. The line l_2 can in this constellation reach a_2 and b_2, and because l_1 is left from l_2, it cannot hit any point at all. As first case, consider $\beta < \gamma$ (figure 4): The first point, which l_2 reaches, is a_2. Here, the rotation is stopped and the new edge $a_2 - b_1$ is added to the triangulation. The rotation center in A is moved to a_2. If $\beta > \gamma$, the first reached point is b_2. The new

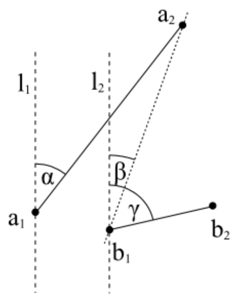

Figure 4: *Triangulating annuli: First case* - $\beta > 0$.

edge $a_1 - b_2$ is added and the rotation center in B is moved to b_2. The last possibility here is $\beta = \gamma$; l_2 hits a_2 and b_2 at the same time. As new edges are $a_1 - b_2$ and $a_2 - b_2$ added and the center of rotation is moved for both lines.

This process is repeated until both startpoints are reached again. In figure 5, a summary of the algorithm is given.

3. Evaluation approach

To assess the quality of the onion peeling triangulation algorithm (OPT) we generated two types of randomly distributed point sets using GNU Octave. The first is a uniform distribution of points within a rectangular area. In the second data set the point density follows a normal distribution from the center to the periphery. We created point sets with varying

Algorithm `TriangulateAnnulus`:
Let A and B be the points of the outer and the inner hull, respectively and T the triangulation of the annulus.

1. Choose a point $a_1 \in A$ and its neighbor $b_1 \in B$ as pair of start-points.
2. Add $a_1 - b_1$ to T.
3. Create two lines of support as vertical lines through $a - l_1$ and through $b - l_2$.
4. Compute the angles α, β and γ.
5. Add vertices and edges to the triangulation and move the center of rotation for the support lines, depending on the angles:
   - If $\beta \leq 0$:
     - If $\alpha < \gamma$: Add edge (a_{i+1}, b_i), move l_1 to a_{i+1}.
     - If $\alpha > \gamma$: Add edge (a_i, b_{i+1}), move l_2 to b_{i+1}.
     - Else: Add edges (a_{i+1}, b_{i+1}) and (a_{i+1}, b_i), move l_1 to a_{i+1} and l_2 to b_{i+1}.
   - Else:
     - If $\beta < \gamma$: Add edge (a_{i+1}, b_i), move l_1 to a_{i+1}.
     - If $\beta > \gamma$: Add edge (a_i, b_{i+1}), move l_2 to b_{i+1}.
     - Else: Add edges (a_i, b_{i+1}) and (a_{i+1}, b_{i+1}), move l_1 to a_{i+1} and l_2 to b_{i+1}.
6. Repeat from 4., until a_1 and a_2 are reached.

Figure 5: *Algorithm for triangulating annuli.*

number of points for both types of distributions and triangulated the point-set using the above described triangulation algorithm. For comparison we use a Delaunay triangulation of the same dataset using GNU Octave. For quality assessment the Delaunay triangulation (DT) can be considered an ideal standard as it maximizes the minimum angle of all angles of a triangle and tends to preserve a well behaved triangulation.

To quantify the quality of a triangulation, we define a symmetry metric as the ratio between the length of the shortest edge and the length of longest edge in a triangle:

$$SM = \frac{E_{min}}{E_{max}}$$

In a perfect tessellation of a surface the symmetry metric SM for all triangles is 1. However, given a point set representing a particular surface, SM will in practice vary across a range from close to zero (extremely skinny triangles) to one (equilateral triangle). To characterize the properties of a specific triangulation algorithm, we determine the distribution function of SM as approximated by the histogram $H(SM)$ (figure 6) for all triangles in a triangulation of a given set of points.

4. Results and discussion

We performed evaluations for a number of randomized point sets ranging from 100 to 3200 points. There were no significant differences in the graphical results nor in the statistics

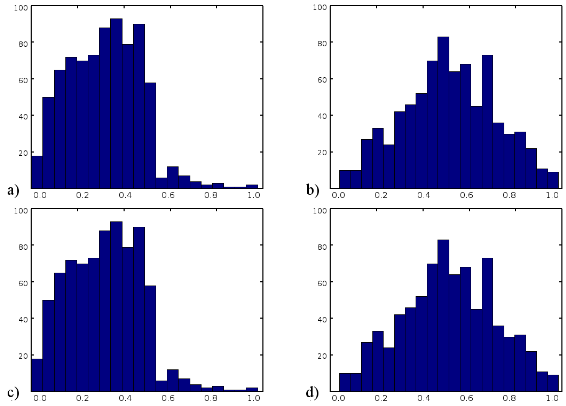

Figure 6: *Distributions of triangle symmetry scores. Upper row: Uniform point distribution, (a) OPT, (b) Delaunay. Lower row: Normal point distribution, (c) OPT, (d) Delaunay.*

for various point set sizes. Therefore we present here the results for the 400-point data set as a representative. Figures 7 and 8 show graphically the results of the triangulations using OPT and DT. DT yields, as expected, to a much more regular triangulation; whereas the onion peeling algorithm tends to produce more skinny triangles with a pronounced circular topology. The distribution of triangle symmetry score, shown in figure 6 suggests a normal distribution for the ideal case of the Delaunay triangulation with a mean triangle symmetry of 0,517 for the uniformly distributed point sets and 0,513 for the normal distributed points, respectively. The symmetry distribution for the onion peeling method is clearly skewed with a bias towards more elongated triangles. The corresponding mean triangle symmetry scores for OPT were 0,310 and 0,314. We had expected this result; it also corresponds to what has been suggested earlier [VK07].

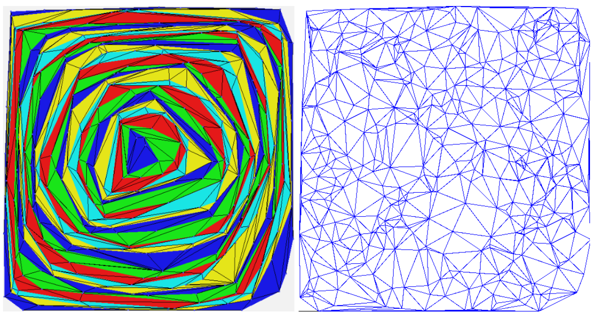

Figure 7: *Uniform point distribution: Left OPT, right DT.*

An interesting finding of our analysis comes from the descriptive statistics of the resulting triangulations. Table 1 summarizes the numbers of triangles and number of strips generated by OPT as well as the number triangles generated

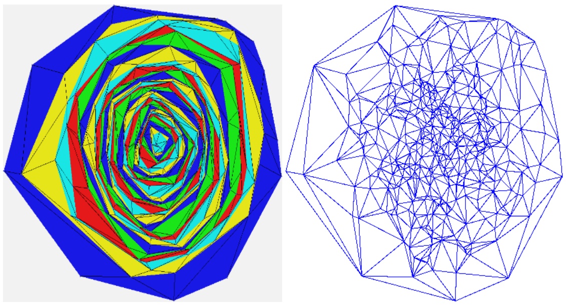

Figure 8: *Normal point distribution: Left OPT, right DT.*

by DT for increasing point set sizes. Although there are significant differences in the shape of the generated triangles, the absolute number of generated triangles is almost identical for both methods regardless of the number of points in the point set (compare row 1 and 4 in table 1). The average number of triangles per triangle strip in OPT (table 1, row 5) ranges from 21 for the smallest point set to 61 in the largest point set. In other words: The average number of vertices to be processed per triangle is as low as 1,09 for the smallest point set and decreases to only 1,03 for the largest point set. We conclude that OPT is a straightforward triangulation method which is certainly favorable as long as triangle shape symmetry is not an issue, considering that the method in itself generates triangle strips with a low average vertex count per triangle.

Number of points	100	200	400	800	1600	3200
#Triangles OPT	189	380	786	1590	3187	6389
#Strips	9	17	26	40	9	17
#Triangles Delaunay	187	380	783	1583	3179	6377
#Triangles / strip	21	22.35	30.23	39.75	47.57	61.43
#Vertices OPT	207	414	838	1670	3321	6597
#Vertices Delaunay	561	1140	2349	4749	9537	19131
#Vertices / triangle	1.095	1.089	1.066	1.050	1.042	1.033

Table 1: *Descriptive statistics for triangulations.*

Although the OPT method as described and evaluated here is a 2D algorithm, it appears as a promising triangulation method for point set surfaces in 3D, as well. Figure 9 shows a triangulation of the Bunny point set. To that end, surface points have been partitioned in 3D using an octree-based subdivision scheme. Point sets in each octree leaf-node have then been triangulated by projection upon a fitting plane, computed with the help of a PCA. The subsequent triangulation in 2D results in a sequence of point indices, later used for ordering the original points into triangle strips. Our initial experiences with this approach indicate that well-behaved triangulation results can be obtained also for 3D point sets when the number of octree-levels is adapted to the convexity of the point surface in 3D. Our current efforts are directed towards exploring this relationship as well as extending onion peeling to handle non-convex hulls.

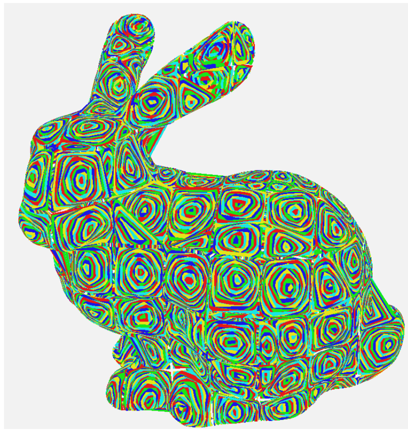

Figure 9: *Bunny data set triangulated with OPT using 4-level octree partitioning.*

References

[AHMS94] ARKIN E., HELD M., MITCHELL J., SKIENA S.: Hamiltonian triangulations for fast rendering. In *Algorithms – ESA '94*, van Leeuwen J., (Ed.), vol. 855 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1994, pp. 36–47. 10.1007/BFb0049395. 1

[BRS05] BOUBEKEUR T., REUTER P., SCHLICK C.: Surfel stripping. In *GRAPHITE '05: Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia* (New York, NY, USA, 2005), ACM, pp. 177–186. 1

[Cha85] CHAZELLE B.: On the convex layers of a planar set. *IEEE Transactions on Information Theory IT-31*, 4 (1985), 509–517. 1

[GE04] GOPI M., EPPSTEIN D.: Single-strip triangulation of manifolds with arbitrary topology. *Computer Graphics Forum (EUROGRAPHICS) 23*, 3 (2004), 371–379. 1

[Tou83] TOUSSAINT G.: Solving geometric problems with the rotating calipers. In *Proceedings of IEEE MELECON '83* (1983), pp. 10–17. 1

[Tou86] TOUSSAINT G.: New results in computational geometry relevant to pattern recognition in practice. In *Pattern Recognition in Practice II* (1986), Gelsema E., Kanal L., (Eds.), pp. 135–146. 1

[VK07] VANĚČEK P., KOLINGEROVÁ I.: Comparison of triangle strips algorithms. *Computers & Graphics 31*, 1 (2007), 100 – 118. 1, 3