New Heuristics for Minimum Weight Triangulation

M. Bartánus, A. Ferko, R. Mag, L. Niepel, T. Plachetka, E. Šikudová

Comenius University, SK-842 15 Bratislava, Slovakia

e-mail: kpgso@fmph.uniba.sk

ABSTRACT

In rendering it is sometimes desirable to compute minimum total light energy mesh. This requires finding the solution for minimum weight triangulation (MWT). We have introduced several new heuristics for MWT, based on original observations. All new algorithms are tested on a set of randomly generated examples. For each example we compute the optimum (for small data sets) using backtrack technique or a reference suboptimum using simulated annealing technique. We compare the new heuristics.

Keywords: minimum weight triangulation, heuristics, brute force algorithm.

1 Introduction

The rendering of wire-frame models can sometimes require the minimum total light energy mesh. This needs finding the minimum weight triangulation MWT. The question whether the construction of MWT is possible in polynomial time has remained open for a long time, [GaJo80], [PrSh85], [Aure91].

There are instances, where the polynomial algorithm [AnCo93] is known. However, the complexity for them is too high, or the input set is very special (e.g. the MWT of the interior of the single polygon can be computed in cubic time, [Gilb79], [Klin80]). This motivates the construction of new heuristics.

The best heuristics known is that by Plaisted and Hong [Plai80], with running time $O(N^2.\log N)$ and achieving the $O(\log N)$ approximation of MWT, but its implementation is too complex. We introduce the practically implementable algorithms. We have developed a simulated annealing research environment, which enables the comparison of algorithms for any input set, and a backtrack program, offering the global optimum for smaller test examples.

The paper is organised as follows. We formulate the problem at the end of this section. Then we introduce some observations and construct three new heuristics in section 2. We compare the heuristics in section 3. The conclusion is given in section 4. The computed results are summarized in the Table 1 in Appendix.

The following figure illustrates the fragment of a picture displaying the top of a skull using 3D Christiansen mesh from a standard renderer. The mesh is projected onto the plane z=0. Evidently, the total edge length can be improved giving better displaying of the same data.

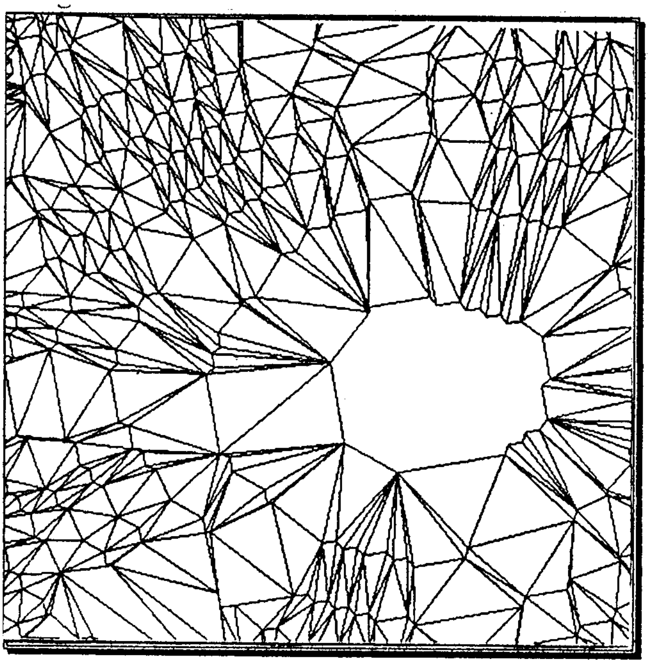

Fig.1: An example of 3D mesh with unsatisfactorily long total edge length.

Thus, the problem is formulated as follows:

Given a set S of N points in the plane in a general position, i. e. no three of them are collinear. Compute the Minimum-Weight Triangulation MWT, which minimizes the total length of the triangulation edges.

2 New Heuristics

In this section we report some observations leading to several new triangulators. There is an exponential number of possible triangulators. A question arises, how to compare these. We will try to evaluate (both theoretically and practically) their quality and complexity. The reference suboptimum for any particular input set is obtained by simulated annealing technique ASA [Čern82] a stochastic optimizing method based on thermodynamic model. The use of ASA for triangulation was designed and implemented in [Boba94]. ASA randomly flips the edges of starting triangulation (DT).

We have also implemented the brute force algorithm based on backtracking, too. Using this algorithm ensures finding the global optimum. Its limitation is its exponential complexity. We were able to compute the results for up to 15 points.

Brute Force Algorithm

Input: S (N distinct points in the plane) Output: MWT(S)

Brute force algorithm:

1. Construct list of all edges
2. Mark all edges in the list as "not considered, not selected, not crossed"
3. Mark all mandatory edges as "considered, selected"
4. Call procedure Backtrack.

Procedure Backtrack

```
If the list does not contain any "not selected, not crossed" edge
  New triangulation found (edges marked as "selected belong to the new triangulation). Save
  it if it is better than the previously found triangulations.
Else
  e := some "not considered, not selected, not crossed" edge
  Mark e as "considered, selected"
  C(e) := set of all "not considered, not selected not crossed" edges crossing e
  Mark all edges in C(e) as "crossed"
  Call procedure Backtrack
  Mark e as "not selected"
  Mark all edges in C(e) as "not crossed"
  For each edge c in C(e)
    Mark c as "considered, selected"
    C(c) := set of all "not considered, not selected not crossed" edges crossing c
    Mark all edges in C(c) as "crossed"
    Call procedure Backtrack
    Mark c as "not considered, not selected"
    Mark all edges in C(c) as "not crossed"
  Mark e as "not considered"
```

Comments: 1. the edge is marked "selected" if it belongs to triangulation in the current level of recursion, otherwise it is marked "not selected".

2. the edge is marked "crossed", if it is crossed by some "selected" edge in the current level of recursion; the edge is marked "not crossed" otherwise.

3. the edge is marked "considered" in two cases: 1. edge is marked "selected", 2. edge is marked "not selected" but some of its crossing edges are marked "selected".

4. The algorithm stems from the following facts: A. Every maximum configuration of edges obtained by successively adding non-crossing edges is a triangulation. B. If an

edge does not belong to a triangulation, then at least one of its crossing edges does.

The algorithm uses these properties to generate all configurations (each configuration considered just once). We optimized the algorithm by not adding an edge when the length of all following configurations was greater than the minimum length so far. This significantly reduces the number of possibilities to be examined, especially if the edges are ordered by some criteria (e.g. length or the weight used in Heuristics 3) so that the first triangulation found is close to the minimum. However, despite the optimizations the time complexity still remains exponential and so this algorithm can only be used for small data sets (our implementation up to 15 points to obtain the result on a SUN SPARC within an hour). Duration times for randomly generated sets of points (measured on a standalone SUN SPARC):

Number of points	1 - 11	12	13	14	15	16 and more
Time (approximately)	1 sec	23 sec	40 sec	20 min	50 min	not measured

2.1 Heuristics 1 (Pairwise Edge Acceptation)

It is well known that MWT can be obtained neither by Delaunay triangulation DT nor by greedy algorithm GRT [PrSh85], [Aure91] although there exist instances of S, for which DT or GRT equals MWT. There is a construction of S with 6 vertices in [Boba94], where the shortest edge does not belong to the MWT. The size of example is possible to reduce. Assume the 4 vertices of any square. There is the locus of points, which are closer to the more distant pair of vertices, at each side of the square. The locus is bounded by the square side and by the border of intersection of two circles centered in the two more distant points with radius equal to the length of the diagonal of given square. We choose the fifth point arbitrarily from this locus. Thus the side of the square becomes a new interior edge in the convex hull of the 5 points. This is the shortest interior edge.

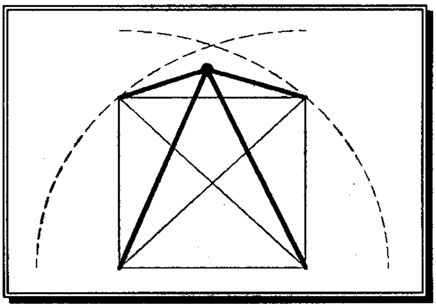

Fig. 2: Minimal test example: optimum is not GRT.

The GRT accepts this edge, but the MWT consists of the 2 diagonals from the added fifth vertex. Evidently, this property cannot be constructed by a smaller set of points. This property leads to the construction of a heuristics for MWT.

Heuristics 1: Pairwise acceptation of interior edges of S.
Input: S (N distinct points in the plane) Output: Approximation of MWT

1. Accept all K edges of CH(S) into the list of accepted edges.
2. Select the current edge (one of accepted ones).
3. Sort all possible pairs of new edges, joining (without intersection with any accepted edge) the endpoints of the current edge with particular point, and identify the minimum length. (If there are more than one such points, select the "middle" one, closest to the axis of current edge.)
4. Accept the pair of edges with minimal length into the list of accepted edges.
5. Remove the current edge from the list of accepted edges and if it is not contained in the output file write it to the output file.
6. If the size of output file is less than 3N-3-K, go to 2.
7. Stop.

Since there are 3 to N extremal points in each S, in step 1 we accept 3 to N edges which must belong to any triangulation. We iteratively accept the pair of edges in step 4, some edges being repeated, as they occur in two neighbouring triangles. The repeating occurrence of them in output file is prevented by step 5. The halting criterion follows from combinatorial analysis of triangulation. This guarantees the result to be a triangulation, if no further edge can be added without intersection with any preceding edge. All this proves that the algorithm identifies a triangulation with linear memory consumption and cubic time. The time complexity follows from the analysis of step 3.

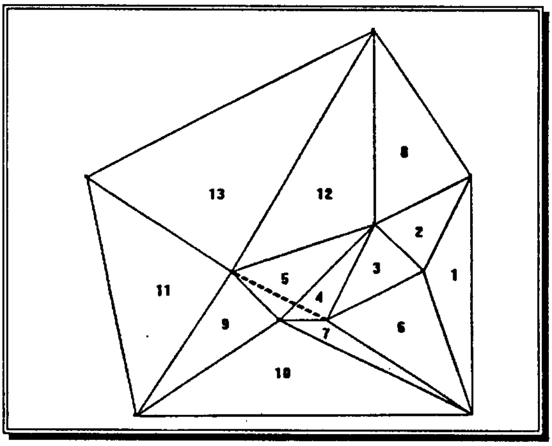

Fig. 3: Illustration of Heuristics 1. The numbers show the ordering of triangle generation.

It seems that the advantage of this algorithm is its easy implementation, but it may cause more than one non-triagulated areas, which must be processed separately (recursively). The modification of the algorithm assuming instead of shortest pair of edges the shortest triangle seems better in some cases. The results for Heuristics 1 are illustrated in Table 1. Optimum is shown by the dashed line in Fig. 3.

2.2 Heuristics 2 (Combining Edges of NNG and ASA)

The Nearest Neighbour Graph NNG has a directed edge if the point is the nearest neighbour of the other. NNG edges belong to DT [ORou94] similarly to the EMST edges, but they are of a somewhat different nature. They can be single or double for a pair of points, and NNG need not be connected. EMST has N-1 undirected edges NNG has N or more directed edges. The Pitteway triangulation [Okab92] is constructed in a similar way - for each triangle every interior point has one of the corners as its nearest neighbour in S. NNG is computable in time O(NlogN).

Heuristics 2: Combining Edges of NNG and ASA

Input: S (N distinct points in the plane) Output: Approximation of MWT

1. Add all double edges of NNG into the list of accepted edges.
2. Remove all crossing edges.
3. Accept all edges which are mandatory now (see section 2.3).
4. Identify remaining edges using DT approach combined with ASA.
5. Stop.

The implementation of this algorithm is in progress. Its innovation for ASA is fixing of a subset of unflippable edges. This follows from the proof that the double NNG edges are allways in MWT, [NiPl96]. The time complexity inherited from NNG is O(NlogN) in step 1, and biquadratic in step 2. The memory consumption is quadratic.

2.3 Heuristics 3 (Edges Mandatory for Each Triangulation)

We have proved, that besides the edges of convex hull some interior edges are mandatory for each triangulation. Extreme edges create a convex hull border. The mandatory edge has at least one endpoint in some interior vertex. Consider the edge which is not intersected by any other possible edge given by a pair of points of S. Evidently, this is true for extreme edges. We show this in two examples (one with one non-extreme edge endpoint, the other with both) in the following figure. Note that there are O(N) mandatory edges possible (Fig. 4b)).

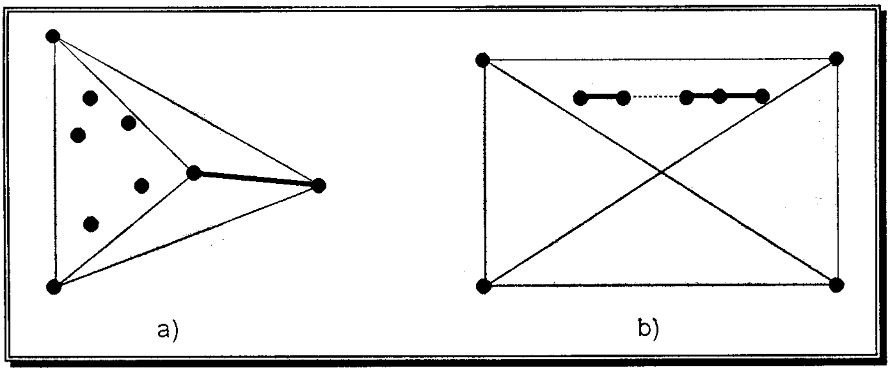

Fig. 4: Mandatory edges for each triangulation: a) one (bold), b) O(N) mandatory edges.

Heuristics 3: Edges Mandatory in Each Triangulation

Input: S (N distinct points in the plane) Output: Approximation of MWT

1. Add all K edges of CH(S) to the list of accepted edges.
2. Accept all non-crossed edges.
3. For each remaining edge e compute the sum of all crossing edge lengths divided by the length of e. Sort the edges by this and choose the edge with maximum value.
4. Remove all the edges crossing the accepted one.
5. If the size of output file is less than 3N-3-K, go to 2.
6. Stop.

In this algorithm we iteratively produce new mandatory edges or edges with the highest "obstacleness", which in turn can produce new mandatory edges, etc. The time complexity is biquadratic, with quadratic memory consumption.

3. Comparison of Results

The simulated annealing, backtrack and triangulation software have been developed at Comenius University. The starting triangulation for ASA is DT, input sets S are generated randomly in a unit square with the size up to 50 points, double points being ignored. The name and the size of randomly generated sets are given in the first column of Table 1. The algorithms are denoted by DT (starting triangulation price), ASA/flips means calculated suboptimum using the reported number of flips, and the name of heuristics - H1, H3. The brute force method is denoted by BFM. Some small input sets were generated manually, e.g. those in Fig. 4.

Table 1 shows that out of 21 BFM-comparable cases there are two where H1 result is longer than H3 result, and 6 where they are equal, but in remaining 13 cases H1 is shorter. In 10 cases H1 is optimal. The comparison of ASA with BFM shows that

ASA in linear to quadratic number of flips computes the optimum. The winning heuristics in this tests is H1, providing often better results than DT.

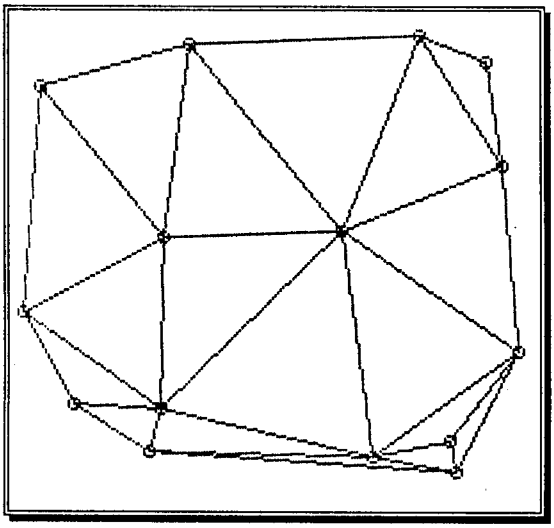

Fig. 5: Illustration of Brute Force Method. The biggest computed optimal solution for 15 points.

The remaining part of Table 1 shows different situation. There is no optimum known. The results of algorithms for bigger input set confirms the advantage of DT. This must be confronted with both GRT and H2 results in future research.

4. Conclusion

In this paper there are given some new heuristics for MWT problem based on original observations. They are compared within a general research framework for evaluation of a quality of a new triangulator. The new heuristics were tested and evaluated as shown in Table 1. The results show that there are several "very good" possible approximations of MWT.

The future research will improve some features of given algorithms and search for the efficiently implementable combination of their advantages. This will be used for creating of a sophisticated 3D mesh generator.

References

[AnCo93] Anagnostou, E. - Corneil, D.: Polynomial-time instances of the Minimum Weight Triangulation problem, Computational Geometry: Theory and Applications 3 (1993), pp. 247-259

[Aure91] Aurenhammer, F. 1991 Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure, ACM Computing Surveys, Vol. 23, No. 3, September 1991, pp. 345 - 405

[Boba94] Bobakova, G. - Niepel, Ľ. - Ferko, A.: On Minimum Weight Triangulation, Proceedings of Spring School on Computer Graphics, Bratislava 1994

[Čern82] Černý, V.: A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm, J. Opt. Th. and Appl. 1982

[Fort93] Fortune, S. A note of Delaunay flips, Pattern Recognition Lettres, special issue Computational Geometry, North-Holland 1993

[GaJo80] Garey, M.R. - Johnson, D.S.: Computers and Intractability, Freeman, San Francisco 1979

[Gilb79] Gilbert, P.D. New Results in Planar Triangulations, M.S. thesis, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois 1979

[Klin80] Klincsek, G.T.: Minimal triangulations of polygonal domains, Ann. Discrete Math. 9 (1980), pp. 121-123

[Laws72] Lawson, C. L. 1972 Generation of a triangular grid with applications to contour plotting, Tech. memorandum 299, California Inst. Tech. Jet Propulsion Lab.

[Ling86] Lingas, A. 1986 The greedy and Delaunay triangulations are not bad in the average case, Inf. Process. Lett. 22, pp. 25-31

[Kirk80] Kirkpatrick, D. G. 1980 A note on Delaunay and optimal triangulations, Inf. Process. Lett. 10, pp. 127-128

[PlNi96] Plachetka, T. - Niepel, Ľ.: Double Edges of ANN are in MWT, Tech. Rep. of Dept. of Computer Graphics and Image Processing No. 96/1, Comenius Univ., Bratislava 1996

[Plai87] Plaisted, D.A. - Hong, J.: A heuristic triangulation algorithm, J. Algorithms 8/87, pp. 405-437

[PrSh85] Preparata, F. Shamos, M.I.: Computational Geometry, An Introduction, Springer 1985

[ORou94] O'Rourke, J.: Computational Geometry in C, Cambridge Univ. Press 1994

[Okab92] Okabe, A. et al.: Spatial Tesselations: Concepts and Applications of Voronoi Diagrams, Wiley 1992

[Sibs77] Sibson, R. Locally equiangular triangulations, Comput. J. 21/77, pp. 243 - 245

Appendix:

Table 1: Comparison of Results.

Pointset / Size	DT	ASA / flips	H1	BFM	H3
6uholnik / 6	5.357683	5.323621 / 153	5.347933	5.323621	5.347933
ela / 8	7.182710	7.049429 / 110	7.049429	7.049429	7.234346
mand6 / 6	6.844679	6.844679 / 0	6.844679	6.844679	7.252423
mand8 / 8	7.084248	7.084248 / 0	7.084248	7.084248	7.152588
mand9 / 9	8.456372	8.072454 / 42	8.264413	8.072454	9.475448
min / 5	5.603550	5.603550 / 0	5.760718	5.603550	5.760718
min2 / 5	5.530820	5.530820 / 0	5.661216	5.530820	5.661216
min3 / 5	5.171512	5.151371 / 8	5.151371	5.151371	5.151371
min6-1 / 6	5.906205	5.906205 / 0	5.906205	5.906205	5.906205
min6-2 / 6	6.136384	5.776718 / 47	5.776718	5.776718	5.776718
min6-3 / 6	6.125655	5.665825 / 20	5.665825	5.665825	5.665825
min6-4 / 6	5.908059	5.535303 / 56	5.633103	5.535303	5.535303
miro / 10	7.747139	7.747139 / 0	7.806375	7.747139	8.872314
rand10-1 / 10	7.374082	7.374082 / 0	7.374082	7.374082	8.076024
rand12-1 / 12	9.658102	9.341994 / 271	9.555521	9.341994	10.818047
rand7-2 / 7	6.942306	6.942306 / 0	6.942306	6.942306	7.286409
rand9-1 / 9	7.303678	7.286353 / 95	7.380473	7.286353	7.363147
y10 / 10	8.292241	8.276486 / 157	8.300539	8.276486	8.524198
y11 / 11	9.513794	9.497957 / 103	9.762265	9.497957	10.039366
y12 / 12	9.463387	9.447272 / 167	9.501336	9.447272	10.286146
set20A/ 20	10.524649	10.5123825/ 321	11.054246	---	14.316322
set20B / 20	11.8868389	11.6900118/ 243	12.463131	---	14.558865
set30A / 30	16.0182781	15.849179/ 338	17.268378	---	18.667758
set30B / 30	18.0019692	17.6252384/ 766	19.644581	---	23.084938
set40A / 40	21.4307861	20.7698802 / 872	22.827764	---	29.922951
set40B / 40	19.9496364	19.9412593/ 990	21.532394	---	27.621193
set45A / 45	22.5625934	22.1042613/1253	24.605662	---	29.138564
set45B / 45	20.4152393	20.1053352/1322	22.828334	---	28.900767
set50A / 50	23.8256530	23.7815303/1655	26.770547	---	34.355074
set50B / 50	21.9355258	21.9274368/1587	23.851022	---	30.052524