Review of the paper "Improved Incremental Randomized Delaunay Triangulation" [Devillers, 1993]

Kenneth Rohde Christiansen, David Henrik Strom

Department of Mathematics and Computing Science
Rijksuniversiteit Groningen
Blauwborgje 3
NL-9747 AC Groningen

{k.r.christiansen, d.h.strom}@student.rug.nl

Abstract
In this review we discuss a paper presenting an alternative algorithm for performing Delaunay triangulation of a set of data point. The new algorithm will be presented and we will discuss the data structures needed. We will also shortly tell how the author of the algorithm arguments for its complexity. The essay will also include a review per se as well as the pinpointing of the difficulties encountered and the procedures taken to overcome problems of this kind. The document assumes that the reader has followed the Geometric Algorithms course given at the RUG or at least a similar course given at another research institution.

Keywords: Triangulation, Delaunay, Randomized, Geometric Algorithms, Computer Science.

1 Introduction

The main objective of this paper is to review a scientific paper introducing an incremental randomized Delaunay triangulation algorithm that the authors claim to be better than the other incremental randomized algorithms on realistic datasets.

The importance and complexity of producing fine-grained meshes has been a leading factor in the development of new algorithms for producing Delaunay triangulations. Thus, it is the aim of this review to focus on one such algorithm developed by Olivier Devillers.

Computing the Delaunay triangulation of a given dataset is one of the classical problems in computational geometry and there also exists a great array of algorithms to solve this problem, with all their pros and cons. Some are optimal on worst case data, some are randomized, and some only preform good on random data.

The author of the paper claims that people implementing triangulators often choose the algorithm based on simplicity and practical efficiency, while still having the risk of having bad performance of some specific kinds of data sets. Bearing this in mind, the author presents an algorithm that uses a quite simple data structure, is proved $O(n \log n)$ on any dataset and also has a good performance on realistic datasets.

As taught in the lectures the locating structure was very important for the complexity of the incremental randomized algorithm. We have to insert n points for each of these n points we have to do some work, which was of $O(\log n)$ in the algorithm presented in our course, which was using some kind of tree structure (without mentioning the exact structure the complexity isn't that surprising when you keep in mind that balanced trees have a depth of $\log n$). These algorithms have the problem that they depend on the history of the inserted points, which means that it depends on the insertion order and it also uses a lot of memory, which cannot be controlled according to the author of this paper.

The author mentions two algorithms that has inspirated this work:

The first one [Mulmuley, 1997] proposes a locating structure that is not dependent on the insertion order. The structure consists of levels consisting of random sample points, for which the Delaunay triangulation is computed. These overlapping triangles from each level are then linked to each other which yields a complexity of $O(\log^2 n)$ for each point. The structure is, according to the paper, however, not very simple and it still uses a lot of additional memory.

The second algorithm [Ernst Muecke and Zhu, 1996], uses a very simple data structure to handle triangulations of random data points. The structure only consists of $\sqrt[3]{n}$ points with links to the incident triangles in the Delaunay triangulation. A new point can then easily be located by finding the nearest neighbor, for instance by brute force. The advantage of the algorithm is that it has a complexity of $O(n^{\frac{3}{4}})$ for **evenly** distributed points and the constant is very small. This means that for point sets like these it can compete with algorithms of order $O(n \log n)$, but for other data sets the complexity can be far worse.

1.1 The proposed idea

The proposed idea uses a level structure just as [Mulmuley, 1997], but instead of having complex direct relations between the different levels, it uses a match like [Ernst Muecke and Zhu, 1996] in order to locate the points. Using these simpler relations better allows controlling the memory overhead.

2 The algorithm

In the following S is our data set consisting of n points. The algorithm will compute the Delaunay triangulation of S, called DT_S. The algorithms will also be efficient under insertions and deletions.

2.1 The location structure

The location structure of an incremental Delaunay algorithm is the data structure used to locate vertices and edges of the triangulation. There are various methods for doing this, as for instance the sweep-line and Delaunay-tree approaches dealt with in the Geometrical Algorithms course at the RUG.

The location structure of the suggested algorithm is based on levels and is similar to the Mulmuley location structure. In the level approach the input set S is divided into k levels with some specific characteristics. The levels are said to be decreasing from k to 0, meaning that level i, $0 < i < k$, is a subset of level $i - 1$.

Level k is considered the highest level, and level 0 is the lowest one.

A level i contains all the points in subsets S_{i-1} to S_k as well as extra points. It also the Delaunay triangulation of S_i, DT_S.

$$S = S_0 \supseteq S_1 \supseteq S_2 \supseteq S_3 ... \supseteq S_{k-1} \supseteq S_k \tag{1}$$

In addition, the probability of a point p in S_i also being in S_{i+1} is that of one over a constant α. The constant can be modified to change the distribution of points on different levels, the higher the constant the higher value of k, resulting in more levels. If a point p_i is present at level S_i, but not at level S_{i+1} then the point is said to be a vertex of level i.

$$Prob(p \in S_{i+1} | p \in S_i) = \frac{1}{\alpha} \in]0, 1[. \tag{2}$$

2.1.1 Links

Each vertex at level i has a link to all the Delaunay triangles incident to it at its own level and all levels below it, thus to the triangles in DT_j for $0 \leq j \leq k$.

Additionally every triangle (in DT_i) has a link to its three neighbors (in the same Delaunay triangulation, DT_i).

2.2 Locating a query

The most important part of the algorithm is locating a query. The reason for this is that what separates this algorithm from other randomized Delaunay triangulation algorithms is the locating of a new point.

The idea is that you can locate the point v closest to the query q by first finding the closest point at a higher level; a kind of rougher scale with fewer points, and thus not necessarily containing the actual closest point. When this point is located we know that the closest point will be contained within the circle with the center of q and with the line segment qv' as the diagonal, thus we have reduced our problem to looking at points within this circle.

Using this information we can search for a point closer to the query at the level below, containing more points and thus locate the closest point at each level.

At each level we start at the point located at the level above (recall that $S_{i-1} \supseteq S_i$; thus $v' \in S_i \Rightarrow v' \in S_{i-1}$). Since we haven't located a point at the top level k, we start at a known point. We continue like this for all levels, always using the found point as starting point.

Since each vertex has a link to all triangles containing that vertex at the level below, we can use the following procedure for the located point (or the given start point at level k)

- We look at all triangles incident to the found point and determine which triangle contains the line segment qv'.

- q is not necessarily in this found triangle, but we can find all triangles intersected by the line segment qv' since each triangle has a link to it's neighbors. By doing this we can locate triangle t_i containing q.

- The triangle consisting of points v, v' and v'', will have one point v (maybe even more, but we just need one) that is closest to q. We know that the actual closest point is within the circle with q as center and with qv as diagonal. If v is the closest point and there is only one, we will only have to search for vertices of triangles in the direction though $vv'$and vv''. We check all these triangles and maintains the closest vertex. The paper points out that for a visited triangle containing vertices $ww'w''$ with w as the closest vertex, it is only necessary to check the edges through ww' (and ww'' respectively) if the angle qww' (and qww'' resp.) is smaller than $\frac{\pi}{2}$

2.3 Deletions and insertions

Deleting a point is quite easy because of the simple data structure. In order to delete a point in S, we simply delete the vertex (a point corresponds to a vertex) at each level along with its links. The paper states that there exist algorithms that can do this in $O(d)$ time, where d is the degree of the vertex. Since the average degree is 6 it doesn't matter much if an algorithms with complexity $O(d \log d)$ was used instead.

Inserting a point corresponds to locating the closest point at each level (Locating a query point), since we have to insert the point and update the Delaunay triangulation DT_i at each level i. We determine the level by satisfying the probability function given in section 2.1 (2) and insert the point at the found level j and all levels below. This way we satisfy the (1) in 2.1.

2.4 Proof of worse case time complexity

In the paper the time complexity is proved to be of order $O(n \log n)$. We don't want to go into the actual proof here in the review paper, but instead state the strategy that they used.

The proof is splitted up into 4 lemmas and then a theorem that simply puts the lemmas into one proof. Additionally, there is a corollary of the theorem. These four lemmas (lemma 2 is a lemma used in lemma 3) prove the three parts stated in section 2.2.

The first lemma states that the expected degree of vertex v_i in DT_{i-1} is of order $O(1)$. This is probably not a big surprise since we know that a Delaunay triangulations has vertices of maximal degree of 6.

The second lemma is a lemma that is actually used by lemma 3. It states that the expected number of vertices q in $R_i = S_i \cup \{newpoint\}$ such that $w \in R_i$ belongs to the disk with center q that passes through q's nearest neighbor in $R_{i+1} \subseteq R_i$ is less that 6α.

The third lemma uses the second lemma to show that the expected number of edges of DT_i intersecting segment qv_{i+1} is $O(\alpha)$.

The fourth lemma says that the expected numbers of triangles of DT_i visited during the last search for v_i from triangle t_i is $O(\alpha)$.

The theorem states that the expected cost of inserting the n^{th} point is $O(\alpha \log_\alpha n)$, which easily can be concluded from lemma 1, 3 and 4. All of these lemmas state the cost at a given level is $O(\alpha)$, and since the height of the locating structure is

$\log_\alpha n$ then the total cost for inserting a point is $O(\alpha \log_\alpha n)$. Inserting n points then yield a time complexity of order $O(\alpha n \log_\alpha n)$.

2.5 Comparison

The paper features a detailed comparison between the [Ernst Muecke and Zhu, 1996] algorithm and the presented algorithm. The comparison is done by looking at the cost for the *classical walk* ($c_1(n)$ - cost with one level) which corresponds to inserting a point when there is only one level and then looking at the cost for inserting in a structure with k levels ($c_k(n)$ - cost with k levels). Additionally, the cost of a mixed approach is considered ($c_k^*(n)$ - cost for mixed approach with k levels). This approach uses the [Ernst Muecke and Zhu, 1996] for the first level k and the presented algorithm for the other levels. The conclusion is that $c_1(n)$ becomes better that $c_1^*(n)$ for $n > 40$, $c_2(n)$ better than $c_1(n)$ for $n > 180$ and $c_2(n)$ better than $c_1^*(n)$ for $n > 600$. They conclude that the proposed algorithm is significantly better than [Ernst Muecke and Zhu, 1996] for $n > 10.000$, thus for quite large data sets.

The algorithm is also compared with other algorithms. The comparison is not done so detailed as above, but instead an implementation of the mixed approach is compared with implementations of various other methods. The conclusion was that the proposed algorithm was significantly faster that other incremental algorithms, but around 50% slower than the divide and conquer algorithm, an algorithm considered out of scope for this review.

3 The Paper

3.1 Our approach to the paper

One of the hard parts to understand of the paper were the proofs. They were well organized with nice lemmas, but the proofs of the lemmas were not that easy to understand at first sight. The way we over came this, was to read the proofs carefully again and again and then discuss them with each others. Even some of the lemmas were not that easy to understand and maybe the wording could have been a bit more clear. When you have first understood the lemmas you quickly forget the proofs of them again and you will have to spend some time understanding them again. That is not such a big problem though, as you at that time are quite convinced of their correctness.

3.2 Quality of the paper

In this section we look into the quality aspects of the reviewed paper. In our attempts to determine the quality level we decided to investigate the following quality attributes of the paper:

- Clarity and completeness of the arguments
- Structure of the arguments and proofs, and
- Interpreted value of the algorithm

The improved algorithm provided by Devillers is based on earlier work presented in preceding research papers. The background for the new algorithm is briefly covered in the beginning of the reviewed paper together with references to these papers.

However, perhaps due to its close relation to the preceding Mulmuley algorithm certain central properties of the Devillers algorithm is left more or less ignored in the beginning of the paper, which lead to rather severe initial confusion, particularly concerning the location structure of the algorithm.

As the paper proceeds to present the newer aspects of the algorithm the arguments and proofs become more structured and complete in their descriptions.

This does not mean that all proofs become easy to understand, they clearly require a sufficiently high understanding of mathematical language, since the author aims for generalized proofs in the paper. However, even though many arguments can be difficult to fully understand they are well structured and follow the sort of textbook examples for proofs which is positive.

In our review we have not attempted to confirm or disprove any of the lemmas or theorems given but merely tried to understand the proofs by following the use of variables and utilized theorems.

This is the most difficult aspect of the paper to access, since it is highly subjective. However, there are some strong arguments supporting the value of the algorithm and the paper presenting it.

The algorithm is proven to give very good results on various data sets, including worst case, random and predicted data sets. Further, the data structure of levels is an easy way to keep track of points and with a well tuned α constant the amount of memory for the data structure is kept low. All these properties are of course desirable, but would be worthless if the author could not prove them in the paper. Therefore the author has dedicated a considerable portion of the paper to experimentally prove the algorithm's increased performance compared to previous algorithms. This, we believe, increases the authors credibility and adds to the value of the paper in general.

3.3 What did we gain from the paper

The most obvious gain from the paper is of course understanding of a new algorithm for Delaunay triangulations. In addition to this a number of other important lessons has also been learned. Of these the following has been identified:

- New understanding of the importance of the location structure of an incredimental Delaunay algorithm
- Improved understanding of mathematical writing
- Increased skills in interpreting mathematical proofs and expressions

Since we have only concentrated on reviewing one paper on a particular algorithm the great majority of gained knowledge and understanding is tightly connected to this particular algorithm. It is also a fact that the prior understanding of the subject was differing between the reviewers, meaning different aspects of the paper proved challenging and thus the newly gained understanding varied.

It is, however, an undisputable truth that the particular algorithm reviewed was entirely new to us and as this review is written, the most exciting new knowledge is the understanding of the algorithm itself and its particular properties, such as the location structure and method for adding and removing vertices to the triangulation.

In the course covered on geometric algorithms the subject of localization structure was covered, but we believe that after reading the paper by Devillers and seeing actual comparisons to other localization structures the understanding of its importance in a Delaunay algorithm increased.

As mentioned earlier the prior experience to mathematical writing was dissimilar, but in the least we can say that the skill in interpreting this kind of texts was improved, even if the degree to which it improved may vary. This is also the most general gain we identified during our study, the ability to absorb new knowledge in a formal language is essential in most technical fields of study.

3.4 How could be paper be improved

The paper was generally not that difficult to understand.

One thing that could be improved though, was the explanation of the location of a query. We found this quite hard comprehending at first. Another thing that also confused us was the figure that consisted of three sub figures. The problem is that the section 'Location of a query' includes three sub items, so we immediately assumed each of the pictures to belong to one item, which they didn't and we only figured it out when we had come to the third sub item. This fact made us very confused and we reread the first two sub items several times because of this. In addition, a few more illustrations would also have helped our understanding of the algorithm.

Another setback to our review was the use of k. S_k was the highest level, consisting of the fewest points, and S_0 was the lowest level consisting of the whole input set. This wasn't mentioned directly and it took some time getting that right in our heads.

The analysis of the time complexity was done by proving lemmas for the separate parts of the algorithms and then combining them in a theorem. The individual lemmas were not that easy to grasp at first and required some special attention from us. I am sure that a few pictures could have helped our understanding here, as well as maybe first stating the idea of the proof before doing the actual proof.

The paper also contained a section where the author evaluates the algorithms on practical cases and then tries tunings the α parameter to be optimal on random datasets. The section is quite straight forward, but we would have liked a proof of how many floating point operations that are used for the different parts of the algorithm instead of just stating some numbers that we cannot confirm easily. Finally, the section also deals with the *classical* *walk*, without explaining what it is.

4 Further reading

If there is interest to read more about the algorithm recommend the reading of the actual paper [Devillers, 1993] as well as the two papers [Mulmuley, 1997] and [Ernst Muecke and Zhu, 1996] which will give the reader more background on the algorithms that gave the inspiration for the algorithm presented in the paper. For more background on the whole subject of Delaunay triangulation we recommend reading [Edelsbrunner, 2001].

5 Conclusion

In this section we present our conclusions derived from the study of the algorithm, reading the paper and conducting the reviewing work. Finally, this is what we have concluded:

- We can conclude from the reviewed paper that the presented algorithm provide improved performance compared to many other Delaunay algorithms but still the algorithm referred to as divide and conquer show better performance in almost all aspects. However, since the divide and conquer algorithm is not incremental we believe there are still situations where the use of this new algorithm will be the better choice.

- Concerning the paper itself we conclude that although the general structure and presentation of the algorithm is well formulated and argued, certain improvement, particularly use of figures and more exhaustive descriptions of the proofs and the algorithm itself, would have made the paper that much more readable.

- Our last conclusion is regarding to our approach to reading and digesting the reviewed paper. We found it very important to pay attention to details, small statements that we at first read maybe did not understand. We found it helpful to skim through the whole paper and then return to each uncertain item in the paper and iteratively build up an understanding of the paper. At certain point we also expanded the formulas (e.g. regarding what set a given point is part of).

This finishes our review of the incremental randomized Delaunay algorithm presented by Olivier Devillers. As a final remark we would like to state, that although, the assignment sometimes proved more difficult and challenging than we expected at first, it has at the same time, been rewarding in the insight it has given in state-of-the-art Delaunay triangulation algorithms.

6 Acknowledgements

We want to thank our teacher Mr. Vegter and his assistant Mr. Plantinga for an interesting and worthwhile course.

References

Devillers, O. (1993). Improved incremental randomized delaunay triangulation.

Edelsbrunner, H. (2001). *Geometry and Topology for Mesh Generation*. Cambridge University Press.

Ernst Muecke, I. S. and Zhu, B. (1996). Fast randomized point location without preprocessing in two- and three-dimensional delaunay triangulations. *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, 274–283.

Mulmuley, K. (1997). Randomized multidimensional search trees: Dynamic sampling. *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, 121–131.