Computational Aspects of Triangulations with Bounded Dilation

Wolfgang Mulzer* Paul Seiferth*

Abstract

Let T be a triangulation on a planar point set S. If T has bounded dilation, then the shortest path distance between any two vertices approximates their Euclidean distance. We examine if such triangulations can be used to design efficient algorithms for various geometric problems.

First, we show that given a triangulation with bounded dilation, one can find the closest pair of points in S in linear time on a pointer machine.

Afterwards, we consider an algorithm by Krznaric and Levcopoulos to compute a hierarchical clustering for S in linear time, once the EMST of S is known. We study how their result can be generalized to MSTs of triangulations with bounded dilation. It turns out that their algorithm remains (almost) correct for any such MST. In general, however, the resulting running time might be superlinear. We identify a sufficient condition for a linear time bound and construct a triangulation without this condition as counterexample.

It remains open to identify interesting classes of bounded-dilation triangulations with this property.

1 Introduction

Delaunay triangulations (DT) constitute perhaps the most famous and most well-studied proximity structure. Given a planar point set S, the DT of S encodes many aspects of the distances between the points in S, and it enables us to compute in linear time many other structures on S, such as the Euclidean minimum spanning tree (EMST) and thus the closest pair of points in S, the Gabriel graph, the nearest-neighbor graph, a quadtree, or a well-separated pair decomposition (see, e.g., [5] and the references therein). But what exactly is it that makes DTs so powerful? How much structure is needed in order to represent the proximity information in S?

A very general family of triangulations that includes the DT is given by triangulations of *bounded dilation*. In these triangulations, the shortest path distance between any two vertices approximates their Euclidean distance by a constant factor. Examples of other triangulations with bounded dilation are given by the minimum weight and the greedy triangulation [2]. We would like to explore how strong this information is, compared to the DT, and if it can be exploited algorithmically.

As an introductory example we show that the closest pair of points in S can be found in linear time, once a triangulation with bounded dilation is known.

Afterwards, we consider an algorithm by Krznaric and Levcopoulos (KL) for computing a hierarchical clustering for a planar point set S in linear time, given the Euclidean minimum spanning tree $\mathrm{EMST}(S)$ [3]. In particular, KL use the c-*clustering*: a subset $U \subseteq S$ is called a c-*cluster* for some constant $c \geq 1$ if the distance $d(U, S \setminus U)$ is greater than $c \cdot \mathrm{rdiam}(U)$, where $\mathrm{rdiam}(U)$ is the diameter of the axis-parallel bounding rectangle for U. The set of all c-clusters for S constitutes a laminar family, i.e., two distinct c-clusters are either disjoint or one is a proper subset of the other. Thus, the set of all c-clusters can be naturally represented as a c-*cluster tree* whose nodes correspond to the c-clusters and whose leaves correspond to the points in S. A relaxed version of these trees that is more flexible, but retains the essential properties, are (c_1, c_2)-*cluster trees*, introduced by Mulzer and Löffler [5]: let $1 \leq c_1 \leq c_2$ be constants. We require that every c_2-cluster is represented in the tree, but allow other clusters to be inserted in the hierarchy, as long as they are at least c_1-clusters.

KL presented an algorithm to compute a c-cluster tree for S from $\mathrm{EMST}(S)$ in linear time and showed that c-cluster trees can be used to speed up the computation of various structures for S, e.g. a quadtree for S and the (approximated) single and complete linkage clustering [3,4]. The correctness proof is based on a characterization of c-clusters in terms of the EMST of S. We show that a similar characterization holds for MSTs of triangulations with bounded dilation. This enables us to argue that the KL-algorithm is also correct for such triangulations. To achieve linear running time, KL need a property of the EMST, which unfortunately does not hold for MSTs of general bounded-dilation triangulations. We construct a counterexample to illustrate this issue.

2 Preliminaries and Notation

Let $G = (V, E)$ be a graph and $U \subseteq V$. The *induced subgraph* $G[U]$ on U is the graph on vertex set U that contains exactly those edges from G with both endpoints in U. Furthermore, we define

*Institut für Informatik, Freie Universität Berlin, Germany, {mulzer, pseiferth}@inf.fu-berlin.de. Supported in part by DFG project MU/3501/1.

$I_G(U) := \{uv \in E(G) \mid u \in U \text{ and } v \notin U\}$ as all edges with exactly one endpoint in U.

Let T be a planar triangulation. The *dilation* of two vertices u, v of T is the ratio of the shortest path distance $d_T(u, v)$ between u and v in T and their Euclidean distance. When considering the maximum of these ratios we obtain the *dilation* $\delta(T)$ of T, i.e., $\delta(T) := \max_{u \neq v \in V(T)} d_T(u, v)/|u, v|$.

3 Finding the Closest Pair in Linear Time

To get familiar with the bounded dilation property, we show how to use it to speed up the computation of the closest pair $\mathrm{CP}(S)$ of a point set S. Let T be a triangulation on S and $d := \delta(T)$. Note that the shortest edge xy of T approximates $|\mathrm{CP}(S)|$ by a multiplicative factor of d, and thus we know $|xy|/d \leq |\mathrm{CP}(S)| \leq |xy|$. In order to find $\mathrm{CP}(S)$, we examine all paths of length at most $d|xy|$. This can be done by using Dijkstra's shortest path algorithm for each vertex v of T, but stopping if a path exceeds length $d|xy|$.

Algorithm 1 Computing $\mathrm{CP}(S)$

Closest-pair(T,d)
Find the shortest edge xy in $E(T)$.
Delete all edges in $E(T)$ with length $> d \cdot |xy|$.
Closest pair $\{p, q\} \leftarrow \{x, y\}$
for all $w \in V(T)$ **do**
    Use Dijkstra's algorithm to find the set of vertices C that are connected with w through a path of length at most $d \cdot |xy|$.
    for all $v \in C$ **do**
        if $|wv| < |pq|$ **then**
            $\{p, q\} \leftarrow \{w, v\}$
        end if
    end for
end for
return $\{p, q\}$

Theorem 1 *Given a bounded-dilation triangulation T, Algorithm 1 computes $\mathrm{CP}(S)$ in linear time.*

Proof. Let xy be the shortest edge of T. The correctness follows from the fact that we examine for each vertex v all paths of length less than $d \cdot |xy|$. By the dilation property we must encounter the closest pair.

For the running time we argue that there are only a constant number of edges with length less than $d \cdot |xy|$ incident to any vertex. Let $v \in S$ and let D_v be the disk centered at v with radius $|xy|/d$. Observe that there is no vertex $w \in S$ lying in D_v or otherwise, by the dilation property, there must be a path between w and v of length less than $|xy|$. But this would contradict the minimality of xy.

Now, let u be an arbitrary vertex and let A be the annulus centered at u with inner radius $|xy|$ and outer radius $d|xy|$. The area of A is $O(|xy|)$. Since every vertex v inside A has an empty disk D_v that covers a constant fraction of A, there can be only a constant number of such v's.

Finally, consider the shortest path tree with root u obtained by the execution Dijkstra's algorithm. Let P be path from the the root in the tree. When stopping the computation of P once its length exceeds $d|xy|$, the tree has depth at most $d + 1$. By the above discussion every inner node has constant degree. Thus, the time spend for every vertex is some constant dependent on d only and the overall running time is $O(n)$. □

4 Bounded-Dilation Triangulations and (c_1, c_2)-cluster Trees

Let S be a planar point set. As mentioned in the introduction, KL describe an algorithm for computing a c-cluster tree for S from $\mathrm{EMST}(S)$ in linear time. We explain how to extend this algorithm to triangulations with bounded dilation. However, we will only be able to obtain a (c_1, c_2)-cluster tree, which is slightly weaker, though sufficient for all practical purposes. But first, we give some idea of how the KL-algorithm works: the key insight lies in the following characterization of c-clusters in terms of the EMST [3, Obs. 5].

Observation 2 *Let S be a planar point set and $G = \mathrm{EMST}(S)$. A subset $U \subseteq S$ is a c-cluster if and only if $G[U]$ is connected and all edges in $I_G(U)$ have length greater than $c \cdot \mathrm{rdiam}(U)$.*

We explain how KL use Observation 2 to find for a given vertex v the smallest c-cluster U that contains it, i.e., the parent of v in the c-cluster tree. For this, we start at v and explore the EMST until we find an appropriate subgraph that fulfills Observation 2.

Algorithm 2 Computing the parent c-cluster for v.

ParentCluster(G, v):
Set $U \leftarrow \{v\}$
Queue $Q \leftarrow$ {shortest edge incident to v}
Set $P \leftarrow$ {edges incident to v that are not in Q}
Set $D \leftarrow \{v\}$
while $Q \neq \emptyset$ **do**
    remove the first edge uw from Q (with $u \in U$)
    add w to U
    update the xy-extremes in D
    add each edge wz (except for wu) to P
    move edges in P of length $< c \cdot \mathrm{rdiam}(U)$ to Q
end while
return c-cluster U

Initially, the set U contains only the vertex v, and the algorithm adds to U the closest neighbor w of v in the EMST G. Then, as long as $I_G(U)$ contains an edge uw with $|uw| < c \cdot \mathrm{rdiam}(U)$ (and $u \in U$), the endpoint w is added to U. Afterwards, all edges in $I_G(U)$ have length at least $c \cdot \mathrm{rdiam}(U)$, so Observation 2 guarantees that U is a c-cluster. Note that $I_G(U)$ is represented by P and Q, where Q contains the short edges in $I_G(U)$ and P the long edges.

By extending Algorithm 2, we can obtain the whole c-cluster tree: every time we find a c-cluster U, the induced subgraph $G[U]$ is contracted to obtain a smaller graph G'. Algorithm 2 is then applied to G'. We must make sure that all child clusters of U are identified before the contraction. This is achieved by an appropriate recursion whenever we try to extend U by a vertex that belongs to some child cluster of U (this can be detected efficiently). See [3] for details.

We now show that Observation 2 also holds for general triangulations with bounded dilation, albeit with a slightly weaker conclusion. More precisely, the KL-algorithm produces a (c_1, c_2)-cluster tree when applied to such triangulations (see [6] for details).

Lemma 3 *Let T be a triangulation on a planar point set S with constant dilation d. Let $G = \mathrm{MST}(T)$ and $c_2 > d$. Furthermore, let $U \subseteq S$ be a subset of S such that $G[U]$ is connected. If every edge in $I_G(U)$ has length greater than $c_2 \cdot \mathrm{rdiam}(U)$, then U is a c_1-cluster for $c_1 = 2(c_2 - 1)/(d + 1)$.*

Proof. Let $A := \{b \in \mathbb{R}^2 \mid d(U, b) < (c_2 - 1)\mathrm{rdiam}(U)\}$ be the region with distance less than $(c_2 - 1)\mathrm{rdiam}(U)$ from U (marked by the dashed line in Fig. 1). We will show that all vertices in $S \cap A$ have distance at least $c_1 \cdot \mathrm{rdiam}(U)$ from U. Let $a \in S \setminus U$ be a vertex inside A. We claim that a is not incident to U:

Claim 4 *The triangulation T contains no edge between a and U.*

Proof. Suppose there is an edge $va \in E(T)$ with $v \in U$. As $d(U, a) < (c_2 - 1)\mathrm{rdiam}(U)$, the edge va has length less than $c_2 \cdot \mathrm{rdiam}(U)$. Thus, va is not an edge of G, since we assumed that all edges in $I_G(U)$ are longer than $c_2 \cdot \mathrm{rdiam}(U)$. Since G is connected, it contains a path from v to a in G. This path must use an edge $e \in I_G(U)$. Replacing e by va yields a shorter spanning tree, contradicting the minimality of G. ☐

The vertex a is not incident to U in T, so the shortest path P from v to a in T uses an edge of $I_T(U)$. This implies that P must leave A at some point. Since $a \in A$, the path P must reenter A. Let h be the distance between the point where P reenters A and a (see Figure 1). So,

$$d_T(v, a) \geq (c_2 - 1)\,\mathrm{rdiam}(U) + h. \tag{1}$$

By the triangle inequality, we have $(c_2 - 1)\,\mathrm{rdiam}(U) \leq h + |va|$ and therefore

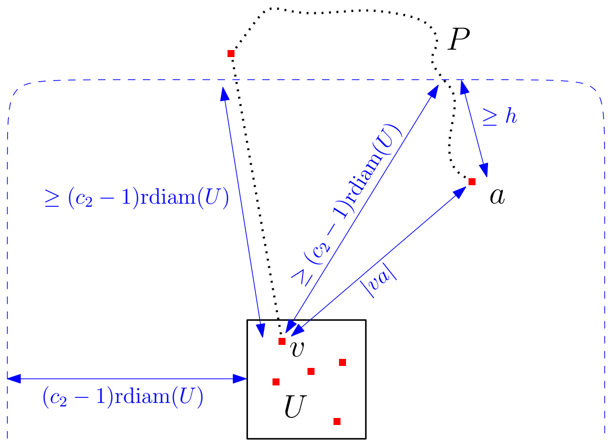

Figure 1: The shortest path from v to a has length at least $2(c_2 - 1)\,\mathrm{rdiam}(U) - |va|$.

$h \geq (c_2 - 1)\,\mathrm{rdiam}(U) - |va|$. Plugging this into (1), we get $d_T(v, a) \geq 2(c_2 - 1)\,\mathrm{rdiam}(U) - |va|$. Since T has dilation d, it follows that

$$d \cdot |va| \geq 2(c_2 - 1)\,\mathrm{rdiam}(U) - |va|$$
$$\Rightarrow |va| \geq (2(c_2 - 1)/(d + 1))\,\mathrm{rdiam}(U).$$

Thus, for every $a \in S \setminus U$, we have $d(U, a) \geq (2(c_2 - 1)/(d + 1))\,\mathrm{rdiam}(U) = c_1 \cdot \mathrm{rdiam}(U)$, so U is a c_1-cluster. ☐

5 Running Time

To argue that their algorithm has linear running time KL used the following generalization of the fact that the EMST of a point set S has constant degree [3]:

Lemma 5 *Let $G = \mathrm{EMST}(S)$, $U \subseteq S$, and $c \geq 1$. If $G[U]$ is connected, then the number of edges in $I_G(U)$ with length greater than $c \cdot \mathrm{rdiam}(U)$ is constant.*

Given the same property for the MSTs of bounded-dilation triangulations, the analysis of the complete adapted algorithm would follow the one for the KL-algorithm. Unfortunately, Lemma 5 does not hold for for such MSTs in general: for every $m \in \mathbb{N}$, we construct a triangulation T_m such that (i) T_m has dilation at most 2; and (ii) the MST of T_m has a vertex of degree m. Since a single vertex can have arbitrarily high degree, this holds also for each subgraph.

Let w be a vertex and set $\alpha := \pi/6$. Choose m vertices $v_1, \ldots, v_m$ in clockwise order such that $\angle v_i w v_{i+1} = \alpha/m$ and $|wv_i| = 3^{i-1}$. We add the edges $wv_1, wv_2, \ldots, wv_m$ to T_m. In order to ensure that these edges belong to $G = \mathrm{MST}(T_m)$, we need some edges that intersect the line segments $v_i v_{i+1}$. Otherwise, these segments would have to be in T_m and also in G. Thus, we place $m - 1$ vertices $u_1, \ldots, u_{m-1}$ on the circle with center w and radius $r = 3^m$ such

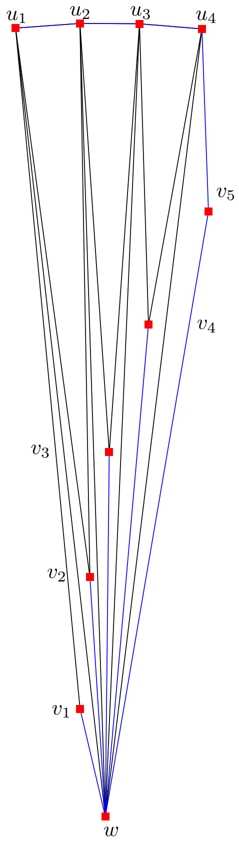

Figure 2: A triangulation with bounded dilation that produces a MST (blue edges) with a vertex of arbitrarily high degree.

that for each i the line wu_i bisects the angle $\angle v_i w v_{i+1}$. To complete T_m, we add the following edges for each $1 \leq i \leq m-1$: (i) wu_i; (ii) $v_i u_i$; (iii) $v_{i+1}u_i$; and (iv) $u_i u_{i+1}$. See Figure 2 shows T_5 (not drawn to scale).

By construction, we have $|v_i u_{i-1}|, |v_i u_i| > |wv_i|$ for all i, so all edges wv_i are in G and w has degree m. It remains to show that T_m has bounded dilation. Indeed, any two nonadjacent vertices a, b in T_m are connected by a path with at most two edges and w as intermediate vertex. We show that the dilation between a and b is at most 2. There are three cases:
Case 1: $a = u_i$ and $b = u_j$ with $i < j$. Let $\beta := \angle u_i w u_j$. Then $|u_i u_j| = 2r\sin(\beta/2)$. The length of the path $u_i, u_{i+1}, \ldots, u_j$ is bounded by the length of the arc between u_i and u_j with center w, i.e., $d_T(u_i, u_j) \leq \beta r$. Thus, the dilation between u_i and u_j is at most $(\beta r)/(2r\sin(\beta/2)) \leq 2$, as $\beta < \pi/6$.
Case 2: $a = v_i$ and $b = v_j$. The largest dilation occurs when v_i and v_j are consecutive, i.e., $j = i+1$. Then $d_T(v_i, v_{i+1}) = |wv_i| + |wv_{i+1}| = 4|wv_i|$, by construction. By the triangle inequality $|v_i v_{i+1}| \geq |wv_{i+1}| - |wv_i| = 2|wv_i|$. The dilation is at most 2.
Case 3: $a = u_i$ and $b = v_j$. The largest dilation occurs for $j = m$ and $i = m-2$. A calculation similar to Case 2 shows that the dilation is at most 2. Thus, T_m has dilation at most 2, as claimed.

6 Conclusion

It remains as an open question, whether there are triangulations with bounded dilation that yield MSTs fulfilling Lemma 5, besides the Delaunay and the greedy triangulation (where the MST is just the EMST). These triangulations can be used to compute a hierarchical clustering of the point set in linear time. Unfortunately, the third popular bounded-dilation triangulation, the minimum weight triangulation, is NP-hard to compute and thus cannot be considered as reasonable input.

A very general candidate would be triangulations that fulfill the *diamond property*, i.e., there exists an angle $\alpha > 0$ such that for any edge e in the triangulation one of the two isosceles triangles with base e and base angle α must be empty. On the one hand, such triangulations have bounded dilation [2], on the other hand they admit a constant-degree subgraph G' that still has bounded dilation (though with a slightly larger constant) and can be found in linear time [1]. Thus, G' is not concerned by the given counterexample.

Finally, note that all steps related to the correctness of the adapted KL-algorithm in Section 4 work with a larger class of, not only triangulations, but even general planar straight-line graphs with bounded dilation. Thus, we can extend our question and ask what kind of planar straight-line graphs yield spanning trees with the necessary properties.

References

[1] P. Bose, M. H. M. Smid, and D. Xu. Delaunay and diamond triangulations contain spanners of bounded degree. *IJCGA*, 19(2):119–140, 2009.

[2] G. Das and D. Joseph. Which triangulations approximate the complete graph? In *Proc. International Symposium on Optimal Algorithms*, pages 168–192. 1989.

[3] D. Krznaric and C. Levcopoulos. Computing hierarchies of clusters from the Euclidean minimum spanning tree in linear time. In *Proc. 15th FSTTCS*, pages 443–455. 1995.

[4] D. Krznaric and C. Levcopoulos. Computing a threaded quadtree from the Delaunay triangulation in linear time. *Nordic J. Comput.*, 5(1):1–18, 1998.

[5] M. Löffler and W. Mulzer. Triangulating the square and squaring the triangle: Quadtrees and Delaunay triangulations are equivalent. *SIAM J. Comput.*, 41(4):941–974, 2012.

[6] P. Seiferth. Computational aspects of triangulations with constant dilation. Master's thesis, FU Berlin, 2012.