Discrete Applied Mathematics 31 (1991) 181–191
North-Holland

Constructing the relative neighborhood graph in 3-dimensional Euclidean space

Jerzy W. Jaromczyk*

Department of Computer Science, University of Kentucky, 915 POT, Lexington, KY 40506-0027, USA

Mirosław Kowaluk

Würzburg Universität, Würzburg, FRG

Received 21 August 1989
Revised 20 February 1990

Abstract

Jaromczyk, J.W. and M. Kowaluk, Constructing the relative neighborhood graph in 3-dimensional Euclidean space, Discrete Applied Mathematics 31 (1991) 181–191.

The relative neighborhood graph for a finite set $S=\{p_1,\ldots,p_N\}$ of points, briefly RNG(S), is defined by the following formation rule: $\overline{p_ip_j}$ is an edge in RNG(S) if and only if for all $p_k\in S-\{p_i,p_j\}$, $\mathrm{dist}(p_i,p_j)\le\max(\mathrm{dist}(p_i,p_k),\mathrm{dist}(p_j,p_k))$.

We show that RNG for point sets in $\mathbb{R}^3$ can be constructed in optimal space and $O(N^2\log N)$ time. Also, combinatorial estimates on the size of RNG in $\mathbb{R}^3$ are given.

1. Introduction

Let $S=\{p_1,\ldots,p_N\}$ be a set of points in $\mathbb{R}^d$. The relative neighborhood graph of S, briefly RNG(S), is a graph with the set of vertices S and the set of edges defined by the following formation rule:

$$\overline{p_ip_j} \text{ is an edge in RNG}(S) \text{ if and only if for all } p_k\in S-\{p_i,p_j\},\ \mathrm{dist}(p_i,p_j)\le\max(\mathrm{dist}(p_i,p_k),\mathrm{dist}(p_j,p_k))$$

where dist denotes the standard Euclidean metric. Further in the paper we will simply use $|\overline{p_ip_j}|$ for $\mathrm{dist}(p_i,p_j)$.

* The work on this paper has been partially supported by the Research Committee Grant of the University of Kentucky and a grant from the Center for Robotics and Manufacturing Systems, University of Kentucky.

0166-218X/91/$03.50 © 1991 — Elsevier Science Publishers B.V. (North-Holland)

Relative neighborhood graphs have been defined by Toussaint [18] as an interesting variant of graphs introduced by Lankford [14]. The immediate interest in these graphs was induced by both their combinatorial flavor and numerous potential applications, e.g. in pattern recognition (see [19]). A number of results on relative neighborhood graphs, their variants, and generalizations have been published; see [7,9-11,13,15-18].

The 2-dimensional case of relative neighborhood graphs has been particularly well studied. The first optimal $O(N \log N)$ algorithm was obtained by Supowit [17] who utilized a clever sweep line approach with sweeping conducted in 6 directions. Lee [15] and O'Rourke [16] addressed 2-dimensional RNG with the metrics L_1 and L_∞. The problem of constructing RNG from the Delaunay triangulation has been investigated by Jaromczyk, Kowaluk and Yao [10], where an optimal $O(N)$ construction has been shown. In fact, they also showed that this result extends to the metric L_p, $1<p<\infty$, and is valid for a wider class of graphs.

On the other hand, the problem of constructing relative neighborhood graphs in higher-dimensional spaces is still unexplored. For a long time no better result than an $O(n^3)$ algorithm presented by Toussaint [18] was known. Apparently the first subcubic algorithm for constructing relative neighborhood graphs in $\mathbb{R}^d$ was given by Supowit [17] for a special class of input points. He designed an $O(N^2)$ algorithm for sets of input points where no three points form an isosceles triangle (or equivalently, all pairwise distances between points in the input set are distinct). Later this result was generalized to the relative neighborhood graphs with the metric L_p, $1<p<\infty$; still, the absence of isosceles triangles was crucial (see Jaromczyk and Kowaluk [9]).

This paper gives an $O(N^2 \log N)$ time and optimal space algorithm for arbitrary point sets in $\mathbb{R}^3$. More specifically we prove the following:

Theorem 1.1. *Given $S=\{p_1,\dots,p_N\}$, $p_i \in \mathbb{R}^3$, $i=1,\dots,N$, the relative neighborhood graph of S can be constructed in $O(N^2 \log N)$ time and optimal $O(\mu_3(S))$ space; $\mu_3(S)$ denotes the size of* RNG(S).

We will also discuss in this paper an estimate for $\mu_3(N)=\max(\mu_3(S)$: S is an N point set in $\mathbb{R}^3)$.

This paper is divided into five sections. The next section gives geometric preliminaries that lay the groundwork for the main algorithm. Section 3 will present our algorithm, Section 4 gives an upper bound on the size of the relative neighborhood graphs in $\mathbb{R}^3$, and finally Section 5 will provide concluding remarks.

2. Geometry of RNG in $\mathbb{R}^3$

The definition of relative neighborhood graphs directly implies a rule that can be used to identify edges that do not belong to RNG. This rule tells that an edge $\overline{p_i p_j}$

does not belong to RNG if and only if there is a point p_k which is closer to both of p_i, p_j than the distance between p_i and p_j. For the purpose of our algorithm, however, it is more convenient to have an elimination rule that is based on relations between the lengths of edges and angles between them rather than on relations between distances.

Such a rule is implied by the following lemma:

Lemma 2.1. $\overline{p_ip_k} \notin \mathrm{RNG}(S)$ *if and only if there exists* p_j *such that* $|\overline{p_ip_j}| < |\overline{p_ip_k}|$ *and*

$$\sphericalangle(\overline{p_ip_j}, \overline{p_ip_k}) < \arccos \frac{|\overline{p_ip_j}|}{2|\overline{p_ip_k}|}.$$

Proof. We will show that under the assumptions $\overline{p_ip_k}$ is the longest edge in the triangle $\triangle p_ip_jp_k$; see Fig. 1. Let $\alpha = \sphericalangle(\overline{p_ip_j}, \overline{p_ip_k})$. By the law of cosines we have

$$\begin{aligned} |\overline{p_jp_k}|^2 &= |\overline{p_ip_k}|^2 + |\overline{p_ip_j}|^2 - 2|\overline{p_ip_j}| \times |\overline{p_ip_k}| \times \cos\alpha \\ &< |\overline{p_ip_k}|^2 + |\overline{p_ip_j}|^2 - 2 \times \frac{|\overline{p_ip_j}|}{2|\overline{p_ip_k}|} \times |\overline{p_ip_j}| \times |\overline{p_ip_k}| \\ &= |\overline{p_ip_k}|^2. \end{aligned}$$

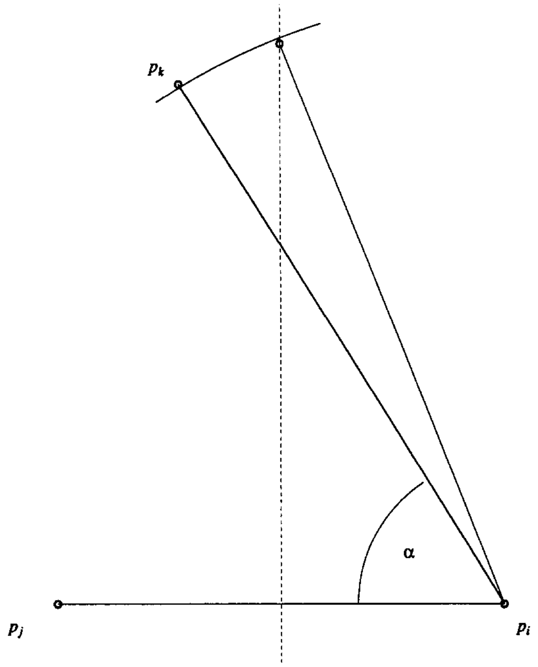

Fig. 1.

Hence $\overline{p_ip_k}$ is the longest edge in $\triangle p_ip_jp_k$ and therefore does not belong to RNG(S). On the other hand, if $\overline{p_ip_k} \notin$ RNG(S), then there exists p_j such that $\overline{p_ip_k}$ is the (strictly) longest edge in $\triangle p_ip_jp_k$. By elementary geometry $\cos\alpha > |\overline{p_ip_j}|/(2|\overline{p_ip_k}|)$ which ends the proof. □

Observe that in particular if the angle between $\overline{p_ip_j}$ and $\overline{p_ip_k}$ ($|\overline{p_ip_k}| > |\overline{p_ip_j}|$) is less than $\frac{1}{3}\pi$, then $\overline{p_ip_k}$ is the longest edge in $\triangle p_ip_jp_k$ and therefore cannot belong to RNG(S). This simple fact which will be very useful later is formulated as the following lemma:

Lemma 2.2. *If* $|\overline{p_ip_j}| < |\overline{p_ip_k}|$ *and* $\sphericalangle(\overline{p_ip_j}, \overline{p_ip_k}) < \frac{1}{3}\pi$, *then* $\overline{p_ip_k} \notin$ RNG(S).

Remark that Lemma 2.1 has the following geometric interpretation: take two points p_i, p_j and the bisecting plane H perpendicular to $\overline{p_ip_j}$. If $|\overline{p_ip_k}| > |\overline{p_ip_j}|$ and $\sphericalangle(\overline{p_ip_j}, \overline{p_ip_k}) < \arccos(|\overline{p_ip_j}|/(2|\overline{p_ip_k}|))$, then $\overline{p_ip_k}$ intersects H. Lemma 2.1 states that no edge $\overline{p_ip_k}$ which is longer than $\overline{p_ip_j}$ and pierces H can belong to the relative neighborhood graph of S.

3. Algorithm to construct RNG in $\mathbb{R}^3$

The algorithm presented in the upcoming sections will consist of two elimination phases responsible for identifying those pairs of points that do not form edges in RNG(S). The first phase, called a *coarse elimination*, will be based on the elimination rule implied by Lemma 2.2 from the previous section. The second phase, called a *fine elimination*, will utilize Lemma 2.1 to eliminate those edges not in RNG(S) that remain after the coarse elimination. For the sake of clarify these two elimination phases, which can be interleaved, will be presented in sequence.

3.1. Coarse elimination

The objective of this phase is to construct a "small" supergraph of RNG(S). This supergraph will have the property that the number of different lengths of edges adjacent to each point is bounded by a constant independent of N. This will be essential for a fast "cleaning-up" process which will be done in the fine elimination. Existence of such a supergraph is guaranteed by Lemma 2.2 which states that the longer of two edges (with a common endpoint) that form an angle not greater than $\frac{1}{3}\pi$ cannot belong to RNG(S).

The coarse elimination will be carried out separately for each point $p_i \in S$. We start with the set $\text{adj}(p_i) = \{\overline{p_ip_j}: j = 1, \ldots, N,\ i \neq j\}$ of all edges having p_i as an endpoint. This set will be subsequently modified by removing some of the edges that do not belong to RNG(S). At each stage all of the (currently) shortest edges in this set are placed in a so-called *orbit*, and then they are removed from the adj(p_i) set.

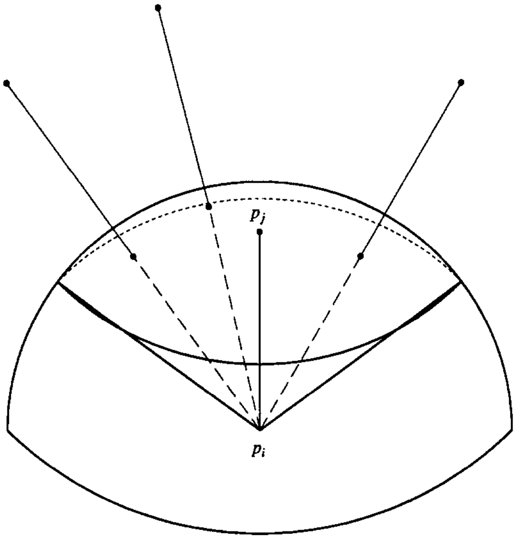

Fig. 2. $\overline{p_i p_j}$ is a central edge.

A representative edge from the orbit is designated as a *central* edge for this orbit and will serve as a basis for further eliminations.

Note that at this moment all edges in adj(p_i) are longer than the central edge. Then those edges that form an angle not greater than $\frac{1}{3}\pi$ with central are removed from the set adj(p_i); they do not belong to RNG(S) by virtue of Lemma 2.2; see Fig. 2.

This process, which forms subsequent orbits and eliminates edges that form with the current central edge an angle less than $\frac{1}{3}\pi$, is repeated until adj(p_i) is empty. Edges eliminated by this process will be placed in the set Coarse_Elim(p_i). Let us note that in fact it is not necessary to store Coarse_Elim(p_i) explicitly; this set is merely used to facilitate a proof of the algorithm correctness. The pseudo-code for the procedure Coarse_Elimination(p_i) is given below.

```
**procedure** Coarse_Elimination($p_i$);
  adj($p_i$) := $\{\overline{p_i p_j}: p_j \in S - \{p_i\}\}$
  number_of_orbits := 0;
  Coarse_Elim($p_i$) := $\emptyset$;
  **while** adj($p_i$) $\neq \emptyset$ **do**
    **begin**
      number_of_orbits := number_of_orbits +1;
      $p_i p_l$ := (one of the) shortest edge in adj($p_i$);
      central($p_i$, number_of_orbits) := $\overline{p_i p_l}$;
      orbit($p_i$, number_of_orbits) := $\{\overline{p_i p_j} \in$ adj($p_i$): $|\overline{p_i p_j}| = |\overline{p_i p_l}|\}$;
      adj($p_i$) := adj($p_i$) – orbit($p_i$, number_of_orbits);
```

```
            Coarse_Elim($p_i$) := Coarse_Elim($p_i$)
                                  $\cup \{\overline{p_i p_j} \in$ adj($p_i$): $\sphericalangle(\overline{p_i p_j}, \overline{p_i p_l}) < \frac{1}{3}\pi\}$;
            adj($p_i$) := adj($p_i$) − Coarse_Elim($p_i$);
        end;
    number_of_orbits($p_i$) := number_of_orbits;
end procedure;
```

The major observation is that the number of the orbits for each point is bounded by a constant independent on N. Indeed, the following lemma holds.

Lemma 3.1. *For all* $p_i \in S$, number_of_orbits$(p_i) \leq c_3$, *where* c_3 *is a constant independent on* N.

Proof. For every central(p_i, j), $j = 1, \ldots,$ number_of_orbits(p_i), let cone$_j$ denote an open circular cone with the apex p_i, symmetric with respect to the line passing through central(p_i, j) and such that the angle at the apex (i.e., the angle in 2-dimensional plane passing through central(p_i, j)) is equal to $\frac{1}{3}\pi$. Let B be a unit ball centered at p_i. Note that $\sphericalangle($central(p_i, l), central$(p_i, j)) \geq \frac{1}{3}\pi$, for $l \neq j$. Hence cone$_l \cap$ cone$_j = \emptyset$ for $l \neq j$. Consequently, the volume of B is not smaller than the sum of the volumes of cone$_j \cap B$, $j = 1, \ldots,$ number_of_orbits(p_i). Since the volume of each cone$_j \cap B$ is fixed the number of cones is bounded by a certain constant c_3. Therefore, the number of central edges, and hence the number of orbits, is bounded by the constant c_3 which is independent on N. □

Lemma 3.2. *For each* $p_i \in S$, Coarse_Elim$(p_i) \cap$ RNG$(S) = \emptyset$.

Proof. This follows from Lemma 2.2 and the way Coarse_Elim(p_i) is constructed. □

Note that at this point we cannot claim that the set Coarse_Elim(p_i) contains all the eliminated edges adjacent to p_i. In fact, the set orbit(p_i, l) may contain some edges that do not belong to RNG(S). However, if all the pairwise distances between points in S were distinct, then the orbits would exactly form RNG(S).

By Lemma 3.1 the number of iterations of the while-loop in the Coarse_Elimination procedure is bounded by a constant c_3; the cost of each iteration is $O(N)$. Hence, we have the following

Lemma 3.3. *The procedure* Coarse_Elimination *constructs the family* orbit$(p_i, l), \ldots,$ orbit$(p_i,$ number_of_orbits$)$ *in* $O(N)$ *time.*

Because the coarse elimination is carried out for each point in S the overall cost of this phase is $O(N^2)$.

Clearly the above presented method is not the only way to construct a supergraph of RNG(S) such that for each point edges adjacent to this point can be grouped into a constant number of orbits consisting of edges of the same length. For example, we can find such a supergraph starting from a general technique developed by Yao [20]. To this end we can think that each point in S is associated with a finite set of vectors (a frame) such that each pair of vectors form an angle of at most $\frac{1}{3}\pi$. Such a frame partitions the space around each point into a finite number of cones. Using Yao's solution to the general geographic neighbor problem, see [20], we can find for all points their nearest neighbors in each of the corresponding cones. The method guarantees that this can be done in a subquadratic time. The edges joining points with such neighbors play a role of our central edges. Clearly, in each cone no edge longer than the corresponding central edge can belong to RNG(S) (because of the angles between vectors in the frame). This straightforward observation can be used to eliminate further edges in the way similar to the coarse elimination. The overall process takes $O(N^2)$ time and produces a supergraph with the desired properties. This graph is, in general, different than one produced by the Coarse_Elimination procedure.

3.2. *Fine elimination*

The family of orbits constructed by the Coarse_Elimination procedure may contain edges that do not belong to RNG(S). The objective of the fine elimination is to discard those edges. For each edge $\overline{p_i p_j}$ in orbit and every shorter edge (than $\overline{p_i p_j}$) the angle between them will be measured and then compared to the threshold angle given in Lemma 2.1. To achieve an efficiency the process of comparing angles will be carried out simultaneously for all edges in a given orbit using an algorithm for the problem of point inclusion in the union of circles.

Before giving a more detailed description of the algorithm let us introduce some notation which will facilitate further presentation. Let V, and W be two sets of edges such that all edges in V are shorter than edges in W. We define $V \blacklozenge W = \{\overline{p_i p_k} \in W\colon \sphericalangle(\overline{p_i p_j}, \overline{p_i p_k}) < \arccos(|\overline{p_i p_j}|/(2|\overline{p_i p_k}|)),\ \overline{p_i p_j} \in V\}$. Intuitively, $V \blacklozenge W$ consists of edges in W that can be ruled out from RNG(S) using the elimination rule implied by Lemma 2.1. We will apply this operation to sets W (second operand) which are orbits, i.e., they consist of edges of the same length. With this restriction in mind, we will show later how this operation may be efficiently implemented. Pseudo-code for the procedure Fine_Elimination(p_i) is presented below.

```
procedure Fine_Elimination($p_i$);
  begin
    Elim($p_i$) := Coarse_Elim($p_i$);
    adj($p_i$) := $\{\overline{p_i p_j}\colon p_j \in S\}$;
    for $l := 2$ to number_of_orbits($p_i$) do
      begin
```

```
            $V := \{\overline{p_i p_j} \in \mathrm{adj}(p_i) \colon |\overline{p_i p_j}| <$ the length of central$(p_i, l)\}$;
            Fine_Elim := $V \blacklozenge$ orbit$(p_i, l)$;
            orbit$(p_i, l)$ := orbit$(p_i, l)$ − Fine_Elim;
            Elim$(p_i)$ := Elim$(p_i) \cup$ Fine_Elim
        end
    end procedure;
```

Observe that this procedure applies the elimination rule of Lemma 2.1 to every edge in orbit(p_i, l). According to the definition of the operation ♦ only edges that are not in RNG(S) are removed from orbit(p_i, l). Therefore, for all $p_i \in S$, Elim$(p_i) \cap$ RNG$(S) = \emptyset$. In addition, since each edge in orbit(p_i, l) is tested against all shortest edges, if $\overline{p_i p_j} \notin$ RNG(S), then $\overline{p_i p_j} \in$ Elim(p_i).

Now, both of the above procedures can be combined into a final algorithm.

```
Main_Algorithm
  begin
    for all $p_i \in S$ do
      begin
        Coarse_Elimination$(p_i)$;
        {removes some edges and partitions the others into orbits}
        Fine_Elimination$(p_i)$;
        {removes from orbits all edges not in RNG$(S)$}
      end;
    $G := \bigcup_{i:=1}^{N} \bigcup_{l=1}^{\text{number\_of\_orbits}(p_i)} \text{orbit}(p_i, l)$
  end;
```

Lemma 3.4. $G =$ RNG(S).

Proof. After the coarse and fine eliminations, for each $p_i \in S$,

$$\bigcup_{l=1}^{\text{number\_of\_orbits}(p_i)} \text{orbit}(p_i, l)$$

consists of edges in adj(p_i) − Elim(p_i). In addition Elim(p_i) contains all edges $\overline{p_i p_j}$ not belonging to RNG(S). □

The above lemma implies that the Main_Algorithm is correct, i.e., it constructs RNG(S). The algorithm is organized in such a fashion that each iteration identifies (using O(N) extra memory) the edges in RNG(S) adjacent to p_i. Therefore the whole process needs no more than O$(N + \mu_3(S))$ memory. A subquadratic upper bound for $\mu_3(S)$ will be given in the next section; $\mu_3(S)$ is the size of RNG(S).

The time complexity of the algorithm depends on the implementation of the ♦ operation. Note that this operation is performed on sets of edges V and W where the second operand contains edges of the same length d. Associate with each edge $\overline{p_ip_j} \in V$ a circular cone with its vertex at p_i, symmetric with respect to $\overline{p_ip_j}$, and with the vertex angle equal to $\arccos(|\overline{p_ip_j}|/(2\times d))$. By Lemma 2.1, $\overline{p_ip_j}$ eliminates all edges in W that are in the interior of this cone. Consider a sphere with the origin at p_i and the radius d. Clearly, $\overline{p_ip_j}$ eliminates these edges in W which have their endpoints on the spherical cup formed by the intersection of the sphere with the cone associated with $\overline{p_ip_j}$. This observation reduces the problem at hand to a point location problem in the union of the spherical cups corresponding to cones associated with the vectors in V. A suitable stereographic projection transforms this problem to $\mathbb{R}^2$ where the problem becomes a point location problem for $O(N)$ points in the union of $O(N)$ circles, i.e., given N circles in the plane, determine if a given query per point p is contained in their union. This type of problem has been studied by several authors. In particular, Imai, Iri and Murota [8] have shown that utilizing the Voronoi diagrams in the Laguerre geometry the problem can be solved in $O(\log N)$ time per query point using $O(N \log N)$ preprocessing time. Aurenhammer [1] has utilized power diagrams to the same effect. A solution to the union of circles problem has also been given by Edelsbrunner and Seidel [6] who employed properties of arrangements of hyperplanes. See also Kedem et al. [12]. From these results we can conclude that the ♦ operation can be implemented as an algorithm that uses $O(N \log N)$ steps. This leads to the conclusion given in Theorem 1.1.

4. Upper bound on the size of RNG in $\mathbb{R}^3$

Let $\mu_d(S)$ denote the number of edges in RNG(S) where S is a finite point set in $\mathbb{R}^d$. Define $\mu_d(N) = \max\{\mu_d(S): |S| = N\}$.

From the well-known inclusion MST(S) $\subseteq$ RNG(S) where MST(S) is a minimal spanning tree on S it follows that $\mu_d(N) \geq N-1$ (see [18]). In the two-dimensional case $\mu_2(N) \leq 3N-6$ since RNG(S) is a subgraph of the Delaunay triangulation of S [18]. Previously no nontrivial upper bound on $\mu_d(N)$ has been known for $d \geq 3$. We will show that $\mu_3(N) < cN^{3/2+\varepsilon}$ where c is a constant and $\varepsilon > 0$. This fact will follow from recent results regarding an upper bound $F_3(N)$ on the number of times a particular distance, say the unit distance, occurs among N points in 3-dimensional Euclidean space. The problem of unit distance graphs was raised by Erdös and there is a large number of papers discussing the size of such graphs. In particular, recent results by Clarkson, Edelsbrunner, Guibas, Sharir and Welzl [5] show that $F_3(N) \leq cN^{3/2+\varepsilon}$, where $\varepsilon > 0$ is an arbitrary small real number. In fact they give this bound in a stronger form of $O(N^{3/2}(\lambda_6(N)/N)^{1/4})$, where λ_6 is related to the complexity of Davenport-Schinzel sequences. For more discussion on $F_3(N)$ and $F_d(N)$ in Euclidean spaces see also Chung [4].

Theorem 4.1. $\mu_3(N) \leq cN^{3/2+\varepsilon}$.

Proof. Consider a set $S = \{p_1, \ldots, p_N\}$ of points in $\mathbb{R}^3$. Let $E_1, \ldots, E_k$ be a partition of the set E of the edges in RNG(S) such that for all $v, w \in E_i$, $|v| = |w|$. Let $S_i \subseteq S$ be a set of endpoints of the edges in E_i. The number of different lengths of the edges in RNG(S) adjacent to p_i is bounded by a constant c_3; see the proof of Lemma 3.1. Therefore each point in S can appear in at most c_3 sets S_i and $|S_1| + \cdots + |S_k| \leq c_3 N$. Since each of E_i consists of edges of the same length, by virtue of the results on $F_3(N)$, we have $|E_i| \leq c\,|S_i|^{3/2+\varepsilon}$. By induction, using the inequality $a^\alpha + x^\alpha \leq (a+x)^\alpha$ for $a, x > 0$, $\alpha \geq 1$ we have

$$|E_1 \cup \cdots \cup E_k| \leq c\,|S_1|^{3/2+\varepsilon} + \cdots + c\,|S_k|^{3/2+\varepsilon} \leq c(|S_1| + \cdots + |S_k|)^{3/2+\varepsilon}$$
$$= \text{const} \times N^{3/2+\varepsilon}. \qquad \square$$

Note that the bound of Theorem 4.1 applies also to the supergraph of RNG constructed by the coarse elimination; see Section 3.1.

We believe that the size of the relative neighborhood graphs in $\mathbb{R}^3$ is smaller than the bound given in Theorem 4.1. On the other hand, a standard example shows that $\mu_4(N) = c \times N^2$. To this end, take $\frac{1}{2}N$ points of the form $(a, b, 0, 0)$ where $a^2 + b^2 = \frac{1}{2}$ and the other $\frac{1}{2}N$ points to be of the form $(0, 0, c, d)$ where $c^2 + d^2 = \frac{1}{2}$. The relative neighborhood graph of this set of points has $c \times N^2$ edges.

5. Concluding remarks

A deterministic algorithm constructing relative neighborhood graphs in $\mathbb{R}^3$ has been given. This algorithm is optimal with respect to the space complexity and its time complexity is $O(N^2 \log N)$. The algorithm uses two elimination phases which are based on a property of relative neighborhood graphs which relates lengths of edges with angles between them. This property links the problem to finding whether a given point is contained in the union of circles.

Also an interesting combinatorial problem related to the upper bound for the number $\mu_3(N)$ of edges in RNG(S) has been discussed. Specifically, we have proven that $\mu_3(N) = O(N^{3/2+\varepsilon})$ and this bound has been derived from the combinatorial complexity of unit distance graphs. It would be interesting to find better bounds.

References

[1] F. Aurenhammer, Improved algorithms for discs and balls using power diagrams, J. Algorithms 9 (1988) 151-161.

[2] D. Avis, P. Erdős and J. Pach, Repeated distances in space, Tech. Rept. SOCS-86.21, McGill University, Montreal (1986).

[3] B. Bollobas, Extremal Graph Theory (Academic Press, London, 1978).

[4] F.R.K. Chung, Sphere-and-point incidence relations in high dimensions with applications to unit distances and furthest-neighbor pairs, Discrete Comput. Geom. 4 (1989) 183–190.

[5] K.L. Clarkson, H. Edelsbrunner, L.J. Guibas, M. Sharir and E. Welzl, Combinatorial complexity bounds for arrangements of curves and surfaces, in: Proceedings 29th Annual Symposium on Foundations of Computer Science (1989) 568–579.

[6] H. Edelsbrunner and R. Seidel, Voronoi diagrams and arrangements, Discrete Comput. Geom. 1 (1986) 25–44.

[7] M. Ichino and J. Sklansky, The relative neighborhood graph for mixed feature variables, Pattern Recognition 18 (1985) 161–167.

[8] H. Imai, M. Iri and K. Murota, Voronoi Diagram in the Laguerre geometry and its applications, SIAM J. Comput. 14 (1985) 93–105.

[9] J.W. Jaromczyk and M. Kowaluk, A note on relative neighborhood graphs, in: Proceedings 3rd Annual Symposium on Computational Geometry, Waterloo (ACM, New York, 1987) 233–241.

[10] J.W. Jaromczyk, M. Kowaluk and F. Yao, An optimal algorithm for constructing β-skeletons in L_p metric, SIAM J. Comput., to appear.

[11] J. Katajainen, Bucketing and filtering in computational geometry, Rept. A46, University of Turku (1987).

[12] K. Kedem, R. Livne, J. Pach and M. Sharir, On the union of Jordan Region and collision-free translational motion amidst polygonal obstacles, Discrete Comput. Geom. 1 (1986) 59–72.

[13] D.G. Kirpatrick and J.D. Radke, A framework for computational morphology, in: G. Toussaint, ed., Computational Geometry (North-Holland, Amsterdam, 1985) 217–248.

[14] P.M. Lankford, Regionalization: theory and alternative algorithms, Geogr. Anal. 1 (1969) 196–212.

[15] D.T. Lee, Relative neighborhood graphs in the L_1-metric, Pattern Recognition 18 (1985) 327–332.

[16] J. O'Rourke, Computing the relative neighborhood graph in the L_1 and L_∞ metrics, Pattern Recognition 8 (1982) 45–55.

[17] K.J. Supowit, The relative neighborhood graph, with an application to minimum spanning trees, J. ACM 30 (1983) 428–448.

[18] G.T. Toussaint, The relative neighborhood graph of a finite planar set, Pattern Recognition 12 (1980) 261–268.

[19] G.T. Toussaint, Pattern recognition and geometric complexity, in: Proceedings 5th International Conference on Pattern Recognition, Miami Beach (1980) 1–24.

[20] A.C.-C. Yao, On constructing minimum spanning trees in k-dimensional spaces and related problems, SIAM J. Discrete Math. 11 (1982) 721–736.