Relative Neighborhood Graphs and Their Relatives

by
Jerzy W. Jaromczyk* and Godfried T. Toussaint†

Abstract

This is a survey of results on neighborhood graphs. The paper discusses properties, bounds on the size, algorithms and variants of the neighborhood graphs. Numerous applications including computational morphology, spatial analysis, pattern classification, and data bases for computer vision are described. A rich bibliography of the subject concludes the paper.

keywords: computational geometry, computational morphology, geometric graphs, neighborhood graphs, spatial analysis.

*Department of Computer Science, University of Kentucky, Lexington, KY 40506-0027
†School of Computer Science, McGill University 3480 University Street, Montreal, Quebec, Canada H3A 2A7

Contents

1 Introduction **3**

2 Definitions **4**

3 Properties **6**

4 Size of neighborhood graphs **7**

5 Algorithms **10**
5.1 2-dimensional space 11
5.2 3-dimensional space 15

6 Variants and special cases **18**

7 Applications **24**
7.1 Morphology and Computer Vision 24
7.2 Geographic Analysis 26
7.3 Pattern Classification 27

8 Conclusion **27**

1 Introduction

A decade has passed since the paper *"The relative neighborhood graph of a finite planar set"* [Tou80b] appeared in print. This anniversary gives a good opportunity to review the results obtained so far and the current state of research on neighborhood graphs.

In fact, during this time the original notion of neighborhood has been generalized in several directions and all of these graphs are now jointly referred to as *proximity graphs* [Tou91]. So much interest has been spawned in this area that in December 1989 an entire conference on proximity graphs took place in Las Cruces, New Mexico [DH91].

The relative neighborhood graph of a finite set of points V, $RNG(V)$, is a prominent representative of the family of graphs which are defined using some concept of neighborliness. For points in a real space R^d the relative neighborhood graph of V is a graph with vertex set V and set of edges $RNG(V)$ which are exactly those pairs (p, q) of points for which $\delta(p, q) \leq \max_{v \in V \setminus \{p,q\}} \{\delta(p, v), \delta(q, v)\}$, where δ denotes the distance between p and q. Figure 1 illustrates a set of points and their relative neighborhood graph.

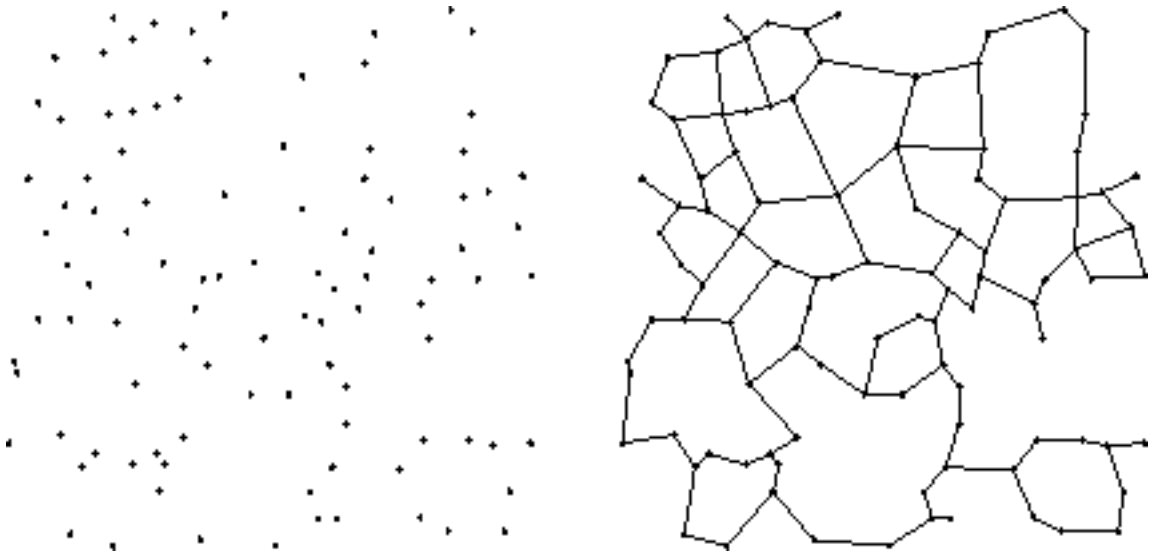

Figure 1: a point set and its *RNG*

The main objective of this paper is to summarize efforts of the last ten years of research on concepts which emerged while studying relative neighborhood graphs.

¿From a mathematical and algorithmic point of view neighborhood graphs fall under the scope of *computational geometry*. More particularly, since they are concerned with extracting shape and structure of point sets, from an application perspective, they form an indispensable tool of *computational morphology*. Neighborhood graphs serve as tools in disciplines where shape and structure are vital. These include visual perception, computer vision and pattern recognition, geography and cartography, and biology, to list a few; for further examples refer to the section on applications and the bibliography which includes numerous references.

There is, however, much beyond purely practical applications which makes neighborhood graphs attractive objects of study. We will demonstrate both practical and theoretical aspects in further sections where we discuss algorithms, properties and applications.

2 Definitions

We will start with a definition of neighborhood graph in a quite general form. This will later enable us to introduce specialized versions more uniformly. Although our later discussion mainly pertains to R^2 and R^3 the definition is valid in any dimension.

Let V be a set of points in R^d. Each (unordered) pair of points $(p, q) \in V \times V$ is associated with a neighborhood $U_{p,q} \subset R^d$. Let $\mathcal{P}$ be a property defined on $\mathcal{U} = \{U_{p,q} : (p, q) \in V \times V\}$. A neighborhood graph $G_{\mathcal{U},\mathcal{P}}(V, E)$ defined by the property $\mathcal{P}$ is a graphs with the set of vertices V and the set of edges E such that $(p, q) \in E$ if and only if $U_{p,q}$ has property $\mathcal{P}$. If (p, q) is an edge, denoted later simply by pq, then we say that q is a neighbor of p (and *vice versa*). For some neighborhood graphs it is more natural to associate neighborhoods with points rather than with pairs of points.

Technically, it is convenient to differentiate between the graph and its geometric realization which is called a *neighborhood skeleton*. The neighborhood skeleton of V is obtained by connecting, with straight line segments, the pairs of points which form edges in the corresponding neighborhood graph. We will use both terms alternatively without, we hope, any confusion. It is worth noting that the term *neighborhood graph* is also used, with a different and unrelated meaning, in graph theory; see [BBD87].

The neighborhood of an edge is usually defined using the concept of distance. In this paper, we will use the metrics $L_p, 1 < p < \infty$, L_1, and L_∞ to measure the distance $\delta(x, y)$ between points $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$ in R^d. The distance in the metric L_p is defined as $\delta_p(x, y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{1/p}$. In L_1 and L_∞ the distance is defined by $\delta_1(x, y) = \sum_{i=1}^{d} |x_i - y_i|$, and $\delta_\infty(x, y) = \max_{1 \le i \le d} |x_i - y_i|$ respectively. Some concepts and results presented in the paper hold for more general metrics. The distance $\delta(p, q)$ will be also called the length of pq. Furthermore, $B(x, r)$ denotes an open sphere centered at x with radius r, i.e., $B(x, r) = \{y : \delta(x, y) < r\}$. A closed sphere is defined as $\overline{B}(x, r) = \{y : \delta(x, y) \le r\}$. In R^2 both a sphere and its boundary will be called, without any confusion, a circle.

Below we will define, within this framework, relative neighborhood graphs, Gabriel graphs, and β-skeletons. Many interesting variants will be given in Section 6.

We begin with an alternative definition of the RNG.

Relative neighborhood graphs: Let $\Lambda_{p,q} = B(p, \delta(p, q)) \cap B(q, \delta(p, q))$; $\Lambda_{p,q}$ is called a *lune*. The relative neighborhood graph of V, the $RNG(V)$, is a neighborhood graph with the set of edges defined as follows:

$$(p, q) \in E \text{ if and only if } \Lambda_{p,q} \cap V = \emptyset.$$

Note that the above definition is equivalent to one given in the introduction.

As a historical digression let us mention that lunes, as intersections of circles, have been the object of intense study since the times of ancient Greeks. Unlike a circle (which cannot be squared

with a straight edge and compass) Hippocrates of Chios (circa 440 B.C.) showed that certain types of lunes could be squared [Dun90]. Furthermore, the type of lune used in the RNG, traditionally referred to as the *Vesica Piscis*, was used with symbolism as a basic design element in the floor plans of gothic cathedrals [Law82].

It is immediate that an edge (p, q) is in the $RNG(V)$ if there is no triangle Δpqv, $v \in V \setminus \{p, q\}$, with pq the strictly longest edge. In Euclidean spaces, relative neighborhood graphs can be equivalently defined by means of angles; we will return to this issue in Section 5.

Note that if we change slightly the definition of the neighborhood by using the intersection of closed spheres (rather than their interiors) we obtain a different class of graphs. In fact such a class was defined by Lankford [Lan69] in 1969 and historically relative neighborhood graphs could be viewed as a modification of this class, see [Tou80b]. As we will point out in later sections this modification leads to different geometric properties.

Gabriel graphs: The neighborhood, called a diameter sphere, is defined as a sphere; $\Gamma_{p,q} = B(\frac{p+q}{2}, \frac{\delta(p,q)}{2})$. The Gabriel graph of V, $GG(V)$, is a neighborhood graph with the set of edges:

$$(p, q) \in E \text{ if and only if } \Gamma_{p,q} \cap V = \emptyset.$$

In the Euclidean space, pq is an edge in $GG(V)$ if there is no triangle Δpqv, $v \in V \setminus \{p, q\}$, with $\angle pvq > \frac{\pi}{2}$. Yet another equivalent definition is that

$$pq \in GG(V) \text{ if } \delta_2(p, q) \leq \min \left\{ \sqrt{{\delta_2}^2(p, s) + {\delta_2}^2(p, q)} \ : s \in V \right\}.$$

Gabriel graphs were introduced by Gabriel and Sokal [GS69] in a context of geographic variation analysis.

β - skeleton: Kirkpatrick and Radke [KR85] defined a parameterized family of neighborhood graphs called β-skeletons. The neighborhood $U_{p,q}(\beta)$ is defined, for any fixed β $(1 \leq \beta < \infty)$ as the intersection of two spheres:

$$U_{p,q}(\beta) = B((1 - \frac{\beta}{2})p + \frac{\beta}{2}q, \frac{\beta}{2}\delta(p, q)) \cap B((1 - \frac{\beta}{2})q + \frac{\beta}{2}p, \frac{\beta}{2}\delta(p, q)).$$

The (lune-based) β-skeleton of V, $G_\beta(V)$, is a neighborhood graph with the set of edges defined as follows:

$$(p, q) \in E \text{ if and only if } U_{p,q}(\beta) \cap V = \emptyset.$$

A useful feature of this parametrized family is its monotonicity with respect to β, i.e. $G_{\beta_1}(V) \subset G_{\beta_2}(V)$ for $\beta_1 > \beta_2$. It is easy to see that β-skeletons contain both relative neighborhood and Gabriel graphs as special cases. Specifically, the $RNG(V) = G_2(V)$ and $GG(V) = G_1(V)$. In fact, as we will see in Section 5 it is possible to design a uniform algorithm for the whole spectrum of β-skeletons for $1 \leq \beta \leq 2$.

β-skeletons have interesting applications to the analysis of interpoint linkages in empirical networks. Kirkpatrick and Radke [KR85] illustrated such an analysis showing examples of road and

airline networks. In particular, a comparison of links in an empirical network with the set of edges in the computed $\beta - skeletons$ for various values of β helps to detect significant patterns.

Lune based β-skeletons can be also defined for $0 < \beta < 1$. To this end the neighborhood of (p, q) is defined as the intersection of two spheres of the radius $\delta(p, q)/2\beta$ which contain p and q in their boundaries. In fact this extension leads to so called *circle-based* β-skeletons. For $\beta \geq 1$ the neighborhood $\mathcal{U}_{p,q}$ of (p, q) is defined as the union of two spheres of the radius $\beta \times \delta(p, q)/2$ passing through $p,\ q$. The points p, q are connected with an edge if the neighborhood is empty. The circle based β-skeleton for $\beta \geq 1$ can be constructed in $O(n \log n)$ time; see [KR85].

3 Properties

Relative neighborhood graphs are related to other prominent geometric structures such as minimum spanning trees (MST) and Delaunay triangulations (DT). The Delaunay triangulation of a set V is defined as the dual graph of the Voronoi diagram of V which is a decomposition of R^d into n cells, n the number of points in V. A point x is in the cell associated with $v \in V$ if for all $w \in V \setminus \{v\}$, $\delta(x, v) < \delta(x, w)$. Two points in the $DT(V)$ are connected with an edge if the boundaries of their Voronoi cells intersect. For definitions and properties of the MST and the DT see for example [PS86, Ede87, Aur91].

Toussaint showed that in the Euclidean plane $MST(V) \subset RNG(V) \subset DT(V)$. It implies, in particular, that the RNG is connected. Figure 3 illustrates the Delaunay triangulation of a point set and nonempty lunes for these of its edges which are not in the RNG.

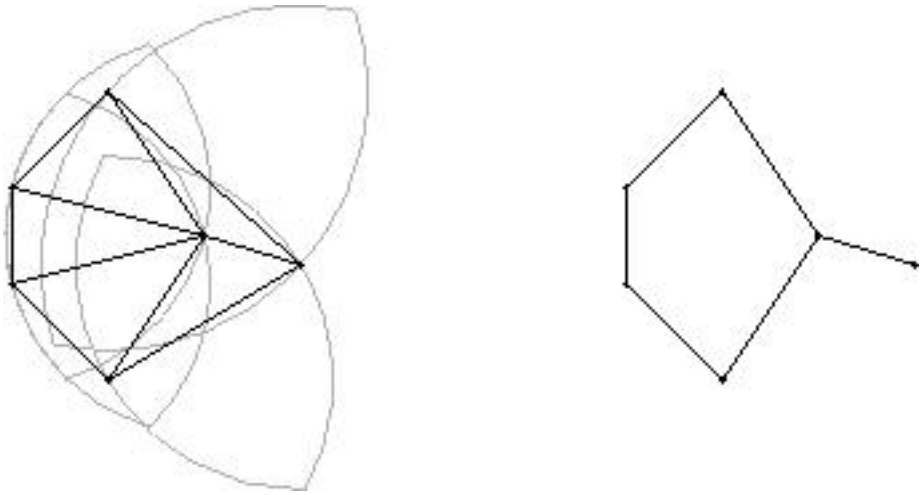

Figure 2: *DT(V)* with four lunes and the *RNG(V)*

These inclusions can be generalized to other metrics and higher dimensions. A crucial step in this direction was made by O'Rourke [O'R82]. Studying the RNG in the L_1 metric he noticed that the dual graph of the Voronoi diagram is not necessarily a supergraph of the RNG in L_1 and L_∞. Instead he proposed a definition of the DT directly in terms of the points of V. Two points in V

are connected by an edge in the DT if there exists a sphere (with respect to the metric at hand) such that its boundary contains these points and no point of V is in the interior of this sphere. This definition is equivalent to the traditional one for the metric L_p in R^2 space; see [Lee80, O'R82]. Here, for the sake of uniqueness of the DT a general position of points is assumed. In our context we say that points are in a general position in R^d if no $d+1$ of them are coplanar (lie on a common $d-1$ flat) and no $d+2$ of them are cospherical (cocircular) with respect to the given metric. In some cases it will be possible to relax the general position assumption.

This definition of the DT allows extension of the result of Toussaint [Tou80b] that the RNG is a subgraph of the DT to all of the metrics L_p, L_1, and L_∞ and to an arbitrary dimension; see [O'R82].

Similar relationships hold for β-graphs in L_p. More specifically, for R^2 and L_p we have $MST(V) \subset G_\beta(V) \subset DT(V)$ $(1 \leq \beta \leq 2)$; see [JKY91] for L_p and [KR85] for L_2. In particular, the relation holds for Gabriel graphs; see also Matula and Sokal [MS84].

As we will see in further sections the fact that the DT is a supergraph of the RNG (and β-graphs) turns out to be very useful in designing efficient algorithms for neighborhood graphs.

4 Size of neighborhood graphs

It is both interesting and important to know lower and upper bounds for the number of edges in the $RNG(V)$. The results of the previous section give immediate bounds in two dimensional space. Since the MST is a subgraph of the RNG and the RNG is a subgraph of the DT we have immediately that $|MST(V)| \leq |RNG(V)| \leq |DT(V)|$. Expressing this in terms of $n = |V|$ we have in L_p that $n-1 \leq |RNG(V)| \leq 3n-6$. The upper bound follows from the planarity of the $DT(V)$. Note that a similar argument works for β-graphs as well. A more detailed analysis gives the tighter bounds of $3n-8$ for the GG (see [MS84]) and $3n-10$, $n \geq 8$, for the RNG (see Urquhart [Urq83]) in the Euclidean plane. The bounds for the RNG and the GG are tight for an infinite number of n. In the metrics L_1 and L_∞ the RNG can have $\Theta(n^2)$ edges; an example has been given by Katajainen [Kat88].

Consider now the case of Euclidean R^d space. As higher dimensional spaces are easier to discuss, we will begin with $d \geq 4$. A discussion on the 3-dimensional case will be deferred.

The first interesting observation is that the relative neighborhood graphs can be dense in R^d, $d > 3$. *The maximum number of edges of the relative neighborhood graph of n points in R^d, $d \geq 4$, is $\Omega(n^2)$.*

Consider a set V in R^4 that contains an even number of points with $n/2$ points of the form $(a, b, 0, 0)$ where $a^2 + b^2 = 1$, and $n/2$ points of the form $(0, 0, c, d)$ where $c^2 + d^2 = 1$. Each pair of points from different groups forms an edge of the $RNG(V)$; this shows that the size of the RNG can be $\Omega(n^2)$. Note that the points are not in general position. By embedding the set V into R^d we obtain this quadratic bound for any $d \geq 4$. Clearly, the same $\Omega(n^2)$ bound holds for all β-graphs. The above construction, based on Lenz's example, is standard and has been used by many authors for various spatial graphs.

The problem of estimating the sizes of the RNG and the GG in 3 dimensional space is much more interesting. Let us start with the RNG; again we focus on Euclidean space. The following

result was the first subquadratic upper bound on the size of the *RNG* (see [JK91]): the maximum number of edges of the relative neighborhood graph of n points in $d \geq 3$ dimensions is $O(n^{3/2+\epsilon})$, for each $\epsilon > 0$.

This result can be established using upper bounds on the size of *unit distance graphs*. The unit distance graph of V is a graph with two points connected by an edge if they are in a particular, say one unit, distance. Recent results by Clarkson *et al.* [CEG$^+$89] show that the size of the unit distance graph of an n point set is bounded by $cn^{3/2+\epsilon}$, where $\epsilon > 0$ 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 the size of unit distance graphs in Euclidean spaces see also Chung [Chu89].

To derive an upper bound decompose the *RNG* into subgraphs of edges of the same length. For each subgraph an upper bound can be established based on the results of the size of unit distance graphs. Furthermore, the number of different lengths of edges in the $RNG(V)$ which are adjacent to any vertex is bounded by a constant independent on V. Hence each vertex participates in a constant number of subgraphs and an upper bound of $O(n^{3/2+\epsilon})$ for the total number of edges in the *RNG* follows. Clearly, any improvement on the upper bound on the size of unit graphs automatically improves the upper bound on the *RNG*.

As a remark in passing let us mention that subgraphs of the *RNG* of edges of the same length have other interesting properties. For example the interior of a pyramid spanned by 3 edges of such a subgraph that share a common endpoint has an empty intersection with the remaining edges of this subgraph. We call this a *non-penetration* property.

Agarwal and Matoušek [AM92] proved an upper bound of $O(n^{4/3})$ using bi-chromatic closest neighbor pairs and an interesting technique of computing a "small" family of well separated pairs which include all edges in the *RNG*; [AESW90]. Let P, Q be two point sets. We say that points $p \in P$, $q \in Q$ form a bi-chromatic closest neighbor pair if $\delta(p,q) = \min_{x \in P, y \in Q} \delta(x,y)$. Furthermore, we say that P and Q are well separated if there exists a pair of cones C, C' with a common apex, a common axis and the angle at the apex less than $\pi/3$, such that $Q \subset C$ and $P \subset C'$. The following holds [AM92]: if P, Q are well separated in R^d then $RNG(P \cup Q)$ has an edge pq, $p \in P, q \in Q$, if and only if (p,q) is bi-chromatic closest neighbor pair. Based on the result of Edelsbrunner and Sharir [ES90], the number of edges in the $RNG(P \cup Q)$ in R^3, P, and Q well separated, is $O(n^{2/3}m^{2/3} + m + n)$, $n = |P|, m = |Q|$. Next, it is possible to cover $V \subset R^3$ by a family of well separated pairs $(P_1, Q_1), \ldots, (P_s, Q_s)$ with two properties: (i) for each $p, q \in V$, there is (P_i, Q_i) with $p \in P_i$, $q \in Q_i$, (ii) $\sum_{i=1}^{s} |P_i| + |Q_i| = O(n \log^2 n)$. Each edge in the *RNG* connects a pair of bi-chromatic neighbors in some well separated pair of this family. Hence counting the number of bi-chromatic neighbor pairs in all of the well separated pairs gives an upper bound on the size of the *RNG*. Concluding: *the maximum number of edges of the relative neighborhood graph of n points in* R^3 *is* $O(n^{4/3})$.

Again, any improvement on the upper bound on the size of bi-chromatic closest neighbor pairs would improve the upper bound on the *RNG*.

We do not know about any nontrivial lower bound on the size of the *RNG* in R^3. For the special case of point sets that do not allow equal distances between pairs of points it is easy to show that the size of the *RNG* is linear. This observation also applies to point sets for which the

number of equidistant neighbors for each vertex is bounded by a common constant; in particular it holds for points in general position. A geometric fact worth noting is that the degree of each vertex of the RNG, if no isosceles triangles are formed, is bounded by a constant. This is also true for $L_p, 1 < p < \infty$ (see for example [JK87]). Therefore $|RNG(V)| \leq c|V|$ for point sets in R^d that do not allow equal distances between pairs of points. Alternatively, the size of relative neighborhood graphs defined by closed lunes is linear.

Size of Gabriel graphs: As we remember, Gabriel graphs are supergraphs of relative neighborhood graphs. It appears that Gabriel graphs can have substantially more edges than the RNG already in R^3. There are point sets in R^3 whose Gabriel graphs have $\Omega(n^2)$ edges; see [Smi89, CEG$^+$90]. Therefore the following result holds: *the maximum number of edges of the Gabriel graph of n points in $d \geq 3$ dimensions is $\Omega(n^2)$.*

An example in [CEG$^+$90] gives a point set V which consists of two groups of $n/2$ points located on two circles. These circles are placed in two orthogonal planes and they pass through each other's center. The points are located close to these centers; this distance can be precisely specified. A construction which uses parabolas, and therefore avoids many cocircular points, was provided in [Smi89]. Consider $2n$ point on two parabolas; n of them on $x = 1 - z^2/4$, $y = 0$ at $z = 0, \pm\epsilon, \pm 2\epsilon, \ldots$ and the remaining n points on the parabola $x = y^2/4 - 1$, $z = 0$, at $y = 0, \pm\epsilon, \pm 2\epsilon, \ldots$ where ϵ a sufficiently small positive number. All the point are on their convex hull. With a small perturbation of the points we can preserve density of the GG while placing the points in general position.

Interestingly, for any set of n points P in $d \geq 3$ dimensions there is a set Q of $O(\sqrt{n}log^{d-1}n)$ points so that the Gabriel graph of $P \cup Q$ has at most $O(n^{3/2}log^{d-1}n)$ edges; see [CEG$^+$90]. Thus by adding extra points we can reduce the size of some Gabriel graphs substantially; furthermore this set of points can be effectively found.

Expected values for proximity graphs: The expected size of Gabriel graphs and relative neighborhood graphs in Euclidean R^d space for various set distributions has been studied, among other graphs, by Devroye [Dev88]. He showed that for all densities $\liminf \mathrm{E}(|GG(V)|)/|V| \geq 2^{d-1}$ ($\mathrm{E}(\gamma)$ is the expected value of a given random variable γ). Furthermore, for most densities $\mathrm{E}(|GG(V)|) \sim 2^{d-1}|V|$. This result extends a similar result by Matula and Sokal [MS84] who demonstrated that for points uniformly distributed in a unit square $\mathrm{E}(|GG(V)|) \sim 2|V|$.

For relative neighborhood graphs the expected size is estimated by $\mathrm{E}(|RNG(V)|)/|V| \geq C_d + o(1)$, where C_d is a constant that depends only on d. In particular, for $d = 2$ the expected number of edges in the $RNG(V)$ is at least $(1.27 + o(1))|V|$.

Several results concerning the expected values for the RNG and the GG, as well as many other geometric graphs, in the unit density Poisson probabilistic model were given in [Smi89].

Let us mention that considerable work has also been done on computing expected values of properties such as the expected length, perimeter, and area of a triangle in the DT. These issues are well covered in Getis and Boots [GB78]; see also [Mil70].

5 Algorithms

A variety of algorithms has been proposed for the RNG in the literature. They are based on ideas developed independently by several researchers. ¿From a retrospective we can identify techniques which are common to several approaches.

A paradigm used in most of the algorithms is a refinement approach. It fits to a framework of *filtering* (for a general treatment and development of this technique and its application to many fundamental geometric problems see Chazelle [Cha86].) Applications of filtering for the RNG have been described in Katajainen [Kat87].

In this approach the RNG is built in stages (usually two) where a sequence of supergraphs with the RNG as a final graph is constructed. Each supergraph is obtained from the previous by eliminating edges which can not belong to the RNG. This elimination, called also a *pruning*, is based either directly on the definition or uses geometric properties of relative neighborhood graphs.

Often such a process applies other fundamental computational problems, as well as nontrivial data structures and general algorithm design techniques.

In fact, common to many algorithms for neighborhood graphs is a general method called the *region approach*; see [GBT84] for various applications. The main idea of the region approach is to divide space in such a way that each point is associated with a finite number of simple regions and the search for the neighbors of the point in these regions reduces to simple queries. One of the most important general references here is Yao's paper [Yao82] presenting subquadratic algorithms for various proximity problems in R^d, e.g., *general geographic neighbors* which are particularly useful in algorithms for neighborhood graphs. In addition, the paper illustrates another powerful technique, *balancing*, i.e., how to trade preprocessing and query costs to minimize the overall processing time.

Not surprisingly, particularly useful for constructing neighborhood graphs are *point location* and *range searching* algorithms. *Point location* consists of identifying the region, in a partition of the geometric space, where the query point is located. *Range-search* is a problem of counting or retrieving all points from a given collection of points which are contained in the query region. A typical query region is a sphere, a lune, a simplex, a halfspace, or a box.

The point location or range-search approach is a natural implication of the way neighborhood graphs are defined; a search of the neighborhood of an edge gives complete information as to whether this edge has to be eliminated. The power of this approach for constructing the RNG was recognized by O'Rourke [O'R82]. He employed an efficient data structure for range queries to construct the RNG in the L_∞ (for arbitrary dimension) and L_1 metrics. The point location approach was also taken by Toussaint, Bhattacharya, and Poulsen [TBP84] to obtain a practically efficient algorithm for computing the Gabriel Graph in all dimensions.

As we will see, the difficulty of the RNG problem will depend on whether the given points are in general position. Recall that in our context we say that points are in general position in R^d if no $(d+1)$ of them belong to a common $(d-1)$ flat and no $(d+2)$ of them are cospherical (cocircular) with respect to a given metric.

5.1 2-dimensional space

A straightforward approach to constructing the RNG is to check for each edge and point if this point is in the neighborhood of the edge. The cost of such a procedure is obviously $\Theta(n^3)$ but this approach does work for any metric or dimension. (In fact, it works for any neighborhoods for which a membership test costs $O(1)$.)

One can immediately reduce this complexity by a factor of n recalling that the Delaunay triangulation (which has $O(n)$ edges) is a supergraph of the relative neighborhood graph of a given point set; this observation led to the first $O(n^2)$ algorithm for the RNG [Tou80b].

Clearly the $DT(V)$ is not the only supergraph which can be effectively utilized for constructing the RNG. A family of such graphs can be obtained using a general approach of *frames* and *narrow regions*; see Yao [Yao82] and Gabow, Bentley and Tarjan [GBT84]. A region associated with a point v is narrow if for $p, s \in V$ that belong to this region we have $\delta(p,s) < \max\{\delta(v,p), \delta(v,s)\}$. Such regions can be obtained, for example, by dividing the space around a given point p into equal sections by 8 lines passing through p. In particular, Katajainen [Kat88] used octant neighbors as a supergraph of the RNG. An octant (geographic) neighbor of p in a region $r(p)$ associated with p is a point of $V \setminus \{p\}$ in $r(p)$ which is closest to p. The octant neighbor graph is obtained by connecting each point of V with its geographic neighbors in all regions. Since an octant neighbor graph can be decomposed into eight planar graphs its size is linear (in L_p, $1 < p < \infty$.) Phasing out edges from this graph by testing them against each point leads to an $O(n^2)$ algorithm.

Using octant neighbors and a range query algorithm, e.g., [Cha86, GBT84, Wil85] for rectangles (intersections of spheres in L_1 and L_∞) Katajainen [Kat88] derived an $O(n \log n + m)$ output sensitive algorithm for the RNG in L_1 and L_∞; m is the output size. To achieve a linear space the range searching is organized accordingly to the batching technique described in [EO85].

For further results regarding geographic neighbors or *the angle restricted nearest neighbor* see also [GS83, Wee89].

Probably the simplest way to divide the plane into regions is to use a square grid. The *cell technique*, see Bentley, Weide, and Yao [BWY80], which leads to a fast expected time algorithm for many closest point problems, has been used by Katajainen, Nevalainen, and Teuhola [KN86, KNT87], see also [TM80], to design a linear expected time algorithm for the RNG in an Euclidean plane. They show that the expected time is linear for the points (in a unit square) generated by a homogeneous Poisson process. Note that the *floor* function is essential to apply the cell technique.

More elaborate elimination techniques: The first $O(n \log n)$ algorithm for the RNG in L_2 was developed by Supowit [Sup83]. Supowit was able to organize the elimination of edges from the $DT(V)$ in $O(n \log n)$ time, matching the cost of building the $DT(V)$. The elimination uses the sweep line technique (see, e.g. [PS86, Ede87]); the sweeping is done in 6 directions (multiples of $\frac{\pi}{3}$). Assume that we sweep from right to left in the direction parallel to a line l. A sweep status structure T maintains *active* points (at the beginning T is empty); E is set to the $DT(V)$. During the sweep any encountered point $p \in V$ is inserted into T. Then, for edges e with the left point p (in the order of increasing angles with respect to l) and for any point $v \in T$ such that the horizontal line passing through v intersects e we check if v eliminates e. If e is eliminated then e is removed

from E, otherwise v is deleted and the plane is swept further. The above process, repeated in the six directions, is able to extract all edges of the $DT(V)$ which do not belong to the $RNG(V)$ and can be implemented to take $O(n \log n)$ time. This gives an $O(n \log n)$ algorithm, which is optimal. In fact Supowit demonstrated a simple linear time reduction of sorting to the RNG in 1-dimensional Euclidean space; the reduction is also valid in L_p.

Using a similar approach Lee [Lee85] has designed an $O(n \log n)$ time algorithm to construct the RNG in L_1 and L_∞ metrics. The starting graph is again the Delaunay triangulation and the assumption that no four points in V are cocircular (in the sense of L_1) is essential.

Construction of the GG in the Euclidean plane is simpler. The first optimal $O(n \log n)$ algorithm was given by Matula and Sokal [MS84], where applications to geographic variation research and clustering were discussed. The algorithm is based on an observation that the Gabriel graph of V contains those edges in the $DT(V)$ which do not intersect their dual Voronoi edges: see Figure 5.1. Since the Voronoi Diagram and $DT(V)$ can be constructed in $O(n \log n)$ time the bound follows. Clearly, the method can be extended to R^d.

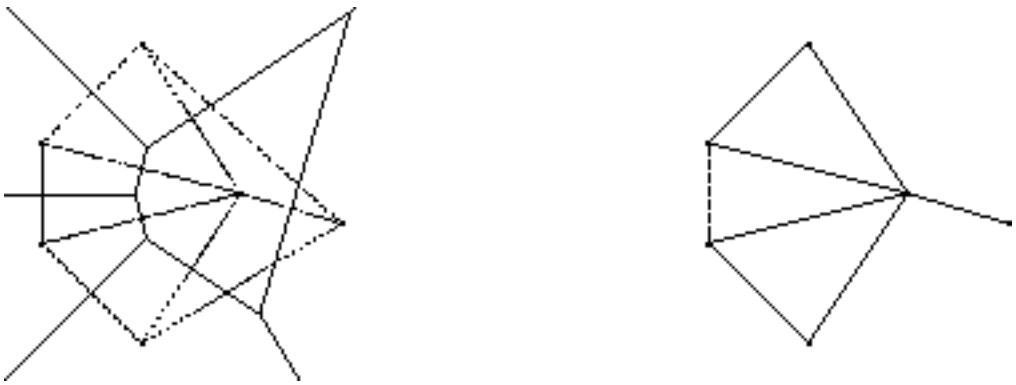

Figure 3: *DT(V)*, the *Voronoi Diagram* of V, and *GG(V)*

Another elimination strategy for the RNG has been proposed in [JK87] and later improved and extended by Jaromczyk, Kowaluk, and Yao [JKY91]. This strategy is based on an interesting structure of edges in the $DT(V)$ that do not belong to the $RNG(V)$. Pick a triangle Δ in the $DT(V)$ and one of its vertices v. Then check if v eliminates the opposite edge of Δ; call this edge e. If yes then analyze if any two edges of the triangle adjacent to Δ along the edge e is eliminated by v. Continue this process until no edge is eliminated by v; the obtained sequence of edges will be called an *elimination path* of v; see Figure 5.1. This process does not necessarily detect all edges eliminated by v. Yet one can prove that each edge not in the RNG belongs to the elimination path for some point in V. If one continues constructing elimination paths for other points it will be noticed that paths that happen to coincide will never split apart. They grow into *elimination trees* (this term is slightly misleading since some elimination paths can have a cycle). This follows from an observation that if a point eliminates an edge of a triangle then it may eliminate only the strictly longer of the two remaining edges of this triangle. This special structure allows an efficient elimination of edges not in the RNG. The elimination paths and elimination trees are built to form so called an *elimination forest* which carries all the information about the RNG. A *union-find*

structure is helpful [GT85].

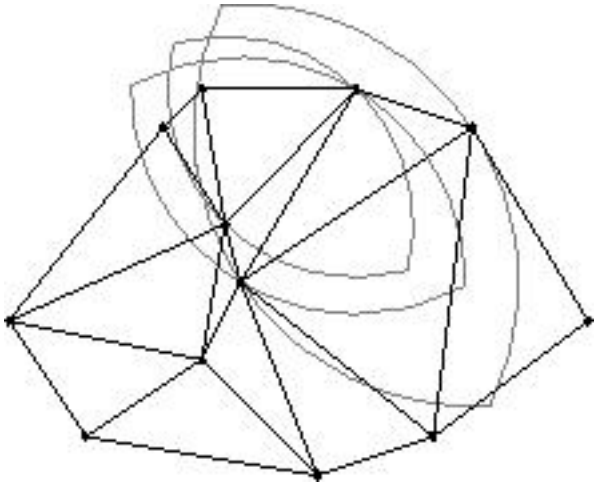

Figure 4: *DT(V)* and lunes showing an elimination path

We can think about this construction as a sweeping guided by the shape of the $DT(V)$ rather than by the coordinate system. This sweeping can be accomplished in time $O(n)$ using the Gabow and Tarjan [GT85] implementation for a special case of *union-find* structure. If we are willing to sacrifice the linear time of this phase to gain simplicity we can use another efficient and simultaneously very simple, almost linear, implementation for *union-find* such as a height ranking with path compression; see [Tar83].

It is interesting to note that the order of constructing elimination paths and composing them into an elimination forest is irrelevant. In general different orders will lead to different elimination forests; nevertheless all edges not in the RNG will always be eliminated. An elimination forest is an important structure and can be useful in other algorithms.

Recently, Hwang [Hwa90] used the idea of elimination paths to construct the RNG in a divide and conquer fashion. Elimination forests are carried over through the levels of recursion.

In view of linear expected time algorithms to construct the $DT(V)$ for some distributions of points, see [Kat87, Dwy88] and [BWY80, Mau84], the idea of elimination forests and results of [JKY91] gives a simple $O(n)$ expected time algorithm for β-graphs.

We will finish this section with a simple algorithm to illustrate the use of the point location method in the elimination phase. This algorithm finds, for a planar point set, all edges in the RNG that are adjacent to a given vertex v. As we will see in the next section, the algorithm has a natural generalization to 3 dimensional space.

We start with the set of all edges adjacent to the vertex v. A currently shortest edge e is found and all edges of the length equal to e are stored. Next all edges longer than e that form with e an angle less than $\frac{\pi}{3}$ are discarded; clearly they do not belong to the RNG. For the remaining (longer than e) edges the same process is repeated until no edges are left. At this moment edges of the same length are grouped into *orbits*. Clearly, by the construction, the number of orbits associated with v is not larger than $(2\pi)/(\frac{\pi}{3}) = 6$.

Not all edges in the orbits have to be in the RNG and the next phase will discard them. A helpful geometric fact is that if the angle between two adjacent edges e and w (w a shorter edge) is smaller than $\alpha_w = arccos(\frac{|w|}{2|e|})$ then e does not belong to the RNG; $|e|$ is fixed for a given orbit. In other words, if for e there is a shorter edge w such that e is contained in the angular sector (cone) with the axis w and the angle equal to $2\alpha_w$ then e is not in the RNG. In this case e is the longest edge in the triangle spanned by e and w. Figure 5.1 illustrates this elimination. The edges on the orbit which intersect the perpendicular bisector of e are not in the RNG; each of them form with w an angle less than α_w.

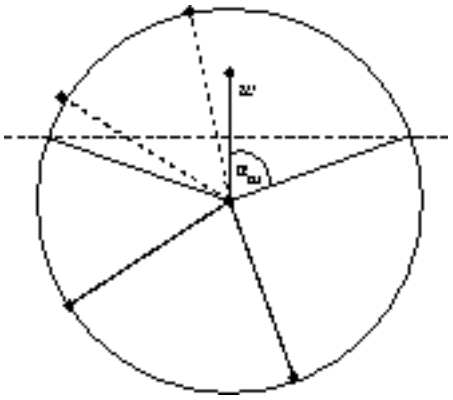

Figure 5: an orbit an its edges eliminated by w

Now the problem reduces to the location of e in the union of angular sectors determined by all edges that are adjacent and shorter than e. After sorting the edges with respect to angles such a union can be found in a linear time. In fact, edges in all orbits can be sorted in an $O(n^2)$ rather than $O(n^2 \log n)$ time. To this end observe that the order of the slopes of lines determined by n points can be found in $O(N^2)$ time; see [OW88]. A simple binary search determines in $O(\log n)$ steps whether e belongs to this union. Since there are at most 6 orbits for a given point and at most n edges the total time to identify all edges in the RNG that are adjacent to a given point v is $O(n \log n)$. Clearly, the same result can be obtained using the Delaunay diagram. We will see in further sections that the above ideas can be extended to the 3 dimensional Euclidean case.

In R^2 the edge location in the union of angular sectors, or equivalently a point location in the union of arcs, can be accomplished faster, in time proportional to n (as suggested by one of the referees.) Observe that each angle of elimination $2\alpha_w > \frac{2\pi}{3}$ and the corresponding arc occupies at least one third of a circle. Let a, b, c be equidistant points on the circle. We can group the arcs with respect to which of a, b, c they contain (ties can be resolved arbitrarily). The union in each group can be determined by finding the extremal arc endpoints in time proportional to the number of arcs. Now each point can be located in the union of all arcs in a constant time. Consequently, all edges in the RNG which are adjacent to a given point can be found in an $O(n)$ time.

For completeness observe that this result yields immediately an $O(n^2)$ algorithm for the RNG in an Euclidean real plane. It is also possible to show that the total number of edges in all the orbits is subquadratic.

We will finish this section mentioning two special cases.

Supowit [Sup83] considered the RNG problem for convex polygons and he provided an optimal $O(n)$ time algorithm. Note that the same result can be established by combining a linear time construction of Delaunay triangulation for points forming a convex hull presented by Aggarwal *et al.* [AGSS87] with a linear time algorithm for constructing the RNG from the DT [JKY91].

A linear time algorithm for computing the RNG of unimodal polygons, not necessarily convex, was designed by Olariu [Ola89].

5.2 3-dimensional space

A straightforward approach gives an $O(n^3)$ time algorithm for the RNG in 3 dimensions, see [Tou80b].

In a manner similar to the 2 dimensional case this time complexity can be reduced by constructing a "small" supergraph of the RNG. The first such construction was given by Supowit [Sup83] for Euclidean spaces. The space around each point of $v \in V$ is partitioned into a number of narrow regions. Recall that a region is narrow if for $p, s \in V$ that belong to the region of v we have $\delta(p,s) \leq \max\{\delta(v,p), \delta(v,s)\}$. The number of regions is independent of n. For points in general position, based on such a partition, it is easy to construct a linear size supergraph of the RNG in $O(n^2)$ time. Note that it can be also done in $o(n^2)$ time using the technique of Yao [Yao82]. Specifically, the only candidate for the RNG in a given region is the edge joining v with the point in this region closest to v (the geographic neighbor). Due to the assumption about general position, in each region there is only a bounded (by a constant) number of such points. Note that the original assumption is that there are no isosceles triangles formed by the points in a given set. The algorithm works also if the number of equal length edges adjacent to each vertex is bounded by a common constant; in particular it holds for points in general position. A straightforward elimination of edges from this graph yields the RNG in total time $O(n^2)$. By virtue of the results by Yao [Yao82] and Gabow, Bentley, and Tarjan [GBT84] Supowit's algorithm extends to L_p metrics.

Elaborating on the above ideas and using balancing Smith [Smi89] showed how the time complexity can be reduced to $O(n^{23/12} \log n)$; general position is assumed.

Using another approach, Jaromczyk and Kowaluk [JK87] demonstrated a simple $O(n^2)$ algorithm for points in general position in L_p (recall that the original assumption was a lack of isosceles triangles). For each point p, from the set of edges adjacent to p we remove iteratively the currently shortest edge pq and eliminate all edges pr such that $\delta(p,r) > \delta(q,r)$. Each such pr is the longest edge in the triangle Δpqr and therefore is not in the RNG. The RNG is the collection of edges which survive such a test, i.e., are never eliminated in this process. When this process terminates the number of edges adjacent to p is bounded by a constant, provided that there are no two equal distances between points of V (or their number is bounded by a constant). It follows from a fact that neither of $pw, pq \in RNG(V)$ is the longest edge in the triangle Δpwq and the angle opposite to the longest edge is greater than some constant $c(d, L_p)$ which depends only on d and the L_p. Hence the number of iterations for each point is proportional to $c(d, L_p) \times n$ and the overall cost of the algorithm is $O(c(d, L_p)n^2)$.

A similar method can be used for constructing the Gabriel graph of V in Euclidean spaces. A

useful observation here is that $pr \notin GG(V)$ if there is q such that $\angle pqr > \pi/2$. Starting with the shortest edge pq adjacent to p we eliminate all edges pr with $\angle pqr > \pi/2$. The process is iterated for all currently shortest non-eliminated edges until no more edges adjacent to p are eliminated. The edges, E, which remain after such elimination are in the GG; the cost associated with each edge in E is proportional to n. It gives an $O(En)$ algorithm for the GG in the Euclidean R^d space. The cost of the algorithm is $\Omega(n^2)$ since the GG is connected and has at least $n-1$ edges. The algorithm doesn't depend on general position of input points. A variant of this algorithm is presented in [Smi89].

A subquadratic algorithm for the RNG of points in d-dimensional spaces has been offered by Sun and Chang [SC91c]. The algorithm constructs the $RNG(V)$ of a n point set in an Euclidean d dimensional space (d is fixed) in time $O(n^{2-a(d)}(\log n)^{1-a(d)})$, where $a(d) = 2^{-(d+1)}$. The points are assumed to be in general position. The geographical nearest neighbor graph is used as a departure point. Recall that the GNG is a linear size supergraph of the RNG, see [Sup83], and it can be constructed in $O(n^{2-a(d)}(\log n)^{1-a(d)})$; see [Yao82]. The next stage, an elimination, uses an arrangement of $(d+1)$ - dimensional hyperplanes. The hyperplanes are the images, in the inverse transformation, of those spheres which bound lunes generated by the edges of the GNG. Next the algorithm identifies lunes which contain a point from V. By well-know properties of the inverse transformation the point location in an arrangement of spheres (lunes) in R^d is translated into location problem in the corresponding arrangement of hyperplanes in R^{d+1}. Su and Chang used an algorithm from Dodge [Dod72] and the balancing technique to preprocess hyperplanes for supporting point location queries.

Using faster existing algorithms for the nearest neighbor search in 3 dimensional spaces they obtained an $O(n^{29/15}\log n)$ time algorithm to construct the RNG of points in R^3 (recall that general position is assumed).

Extending the idea of elimination based on angles between edges Jaromczyk and Kowaluk [JK91] demonstrated an $O(n^2\log n)$ time algorithm for arbitrary point sets in 3-dimensional Euclidean space. The algorithm finds for each vertex v all edges of the RNG adjacent to v. As in the 2 dimensional case *orbits* of edges of the same length are formed (see the previous subsection). The number of orbits for each vertex is bounded by a constant independent of n. This constant is equal to the maximum number of segments with a common endpoint which form angles not less than $\frac{\pi}{3}$.

The orbits associated with v contain all edges in the RNG adjacent to v. However, they can also include extra edges. The size of the graph determined by the union of the orbits for all the vertices is $O(n^{3/2+\epsilon})$; see [JK91] or Section 4. A straightforward elimination leads to an $O(n^{5/2+\epsilon})$ algorithm. We can capitalize, however, on a special form of orbits and perform this elimination faster.

Recall that if the angle between two adjacent edges e and w (w a shorter edge) is smaller than $arccos(\frac{|w|}{2|e|})$ then e does not belong to the RNG. This fact, which we have used to derive a 2-dimensional algorithm, can be utilized in this situation as follows. Consider an orbit of v and let e belong to this orbit. For each edge w with an endpoint v which is shorter than e take an open cone of revolution with its axis containing w and the angle at its apex equal to $2 \times arccos(\frac{|w|}{2|e|})$. Clearly, e is not in the RNG if e is in the union of such cones (with v added).

Alternatively, we can use spherical cups obtained by intersecting the cones with the boundary of

the sphere $B(v, |e|)$ and locate the endpoint of e in the union of these cups. A suitable stereographic projection transforms the problem to a point location in the union of (open) circles. To solve this problem we can use Voronoi diagrams in Laguerre geometry [IIM85] or power diagrams [Aur88]. After $O(n \log n)$ preprocessing such a location can be done in $O(\log n)$ time per point. Hence for each vertex v the cost of eliminating of extra edges from orbits is $O(n \log n)$ which gives in total $O(n^2 \log n)$ time.

Katajainen and Nevalainen [KN87] designed a simple algorithm based on Urquhart [Urq80]. The algorithm works in d dimentional spaces and the metrics $L_1, L_p,\ L_\infty$. Interestingly, its time complexity analysis uses the region approach. In particular, they showed that in R^2 the running time is bounded by $O(n^{5/2})$. Using methods of [JK91] we can in fact prove that this algorithm has $O(n^{5/2+\epsilon})$ time complexity for arbitrary point sets in 3-dimensional Euclidean space.

Noticeable progress in three and higher dimensional spaces has been recently made by Agarwal and Matoušek [AM92]. In the first step a family of well separated pairs of subsets of V is constructed; see Section 4 for the definitions of the well-separated pairs and bi-chromatic neighbors. This family has a property that each RNG edge connects some bi-chromatic closest neighbors in a separated pair of the family. Thererefore, computation of all bi-chromatic closest neighbors for all the points in all the separated pairs gives a supergraph of the RNG. To eliminate from this supergraph edges which are not the $RNG(V)$ an algorithm for a fast point location in an arrangement of lunes is designed. It uses a data structure based on the partitioning scheme of Chazelle *et al.* [CSW90a] and an arrangement of spheres based on Clarkson *et al.* [CSW90b]. If points are in general position then: *for a set V of n points in* R^3 *the $RNG(V)$ can be computed in $O(n^{3/2} + \epsilon)$, for every $\epsilon > 0$.* In R^d, for points in general position the $RNG(V)$ can be constructed in time $O(n^{2(1-\frac{1}{d+1})+\epsilon})$.

It is also possible, see [AM92], to construct the $RNG(V)$ in R^3 in time $O(n^{7/4+\epsilon})$, $\epsilon > 0$, for arbitrary point sets.

In the L_∞ metric the elimination phase can be based on fast point queries in $d-$*rectangles* (intersections of spheres in L_∞). In particular, in R^d the RNG can be constructed in $O(n(\log n)^{d-1})$ time; general position of points is assumed, see [Smi89].

The results discussed in this section for the RNG are summarized in the following table where "big-O" notation is omitted.

dimension	metric	complexity	reference	remarks
R^2	L_p	n^2	[Tou80b, KN86, Kat88]	[KN86] in L_2
		$n \log n$	[Sup83]	in L_2
		n	[JKY91]	when DT is given
	L_∞, L_1	$n^2 \log n$	[O'R82]	
		$n \log n$	[Lee85]	general position
		$n \log n + m$	[Kat88]	m output size
R^3	L_2	$n^{5/2+\epsilon}$	[KN87]	
		$n^2 \log n$	[JK91]	
		$O(n^{29/15} \log n)$	[SC91c]	general position
		$n^{7/4+\epsilon}$	[AM92]	
		$n^{3/2+\epsilon}$	[AM92]	general position
	L_p	n^2	[Sup83, JK87, KN87]	general position
	L_∞	$n^2 \log n$	[O'R82]	general position
		$n(\log n)^2$	[Smi89]	$n(\log n)^{d-1}$ in R^d

The Relative Neighborhood Graphs - Summary of results

6 Variants and special cases

The concept of relative neighborhood as a method to define graphs can be extended and modified in numerous directions. Usually modifications regard either elimination rules or the shape of a neighborhood. This section will discuss some of these interesting variants. The first group of modifications pertains to the relaxation of elimination rules.

k-relative neighborhood graphs: Let $\Lambda_{p,q} = B(p, \delta(p,q)) \cap B(q, \delta(p,q))$, i.e. $U_{p,q}$ is a lune. The edges of the k-relative neighborhood graph of V, $k - RNG(V)$, are defined as follows:

$$(p,q) \in E \text{ if and only if the cardinality of } \Lambda_{p,q} \cap V \text{ is less than } k.$$

Clearly, the 1-$RNG(V)$ is the $RNG(V)$; see Figure 6.

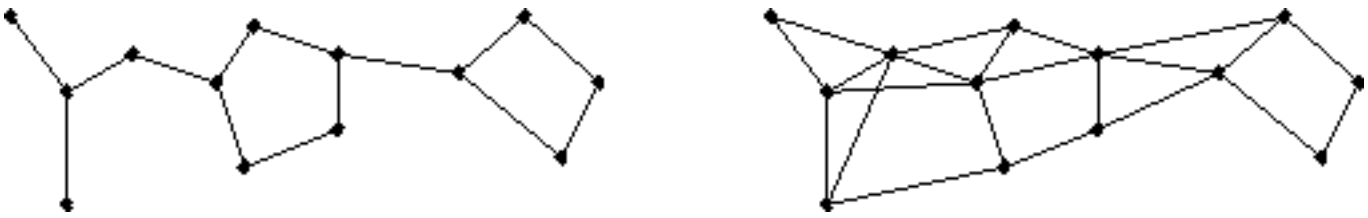

Figure 6: *RNG* and *2-RNG*

Properties and applications of k-$RNG(V)$ in Euclidean 2 dimensional spaces are discussed in [SC91b]. In general, k-RNG graphs in R^2 can contain intersecting edges yet their size is linear with respect to kn.

A simple probabilistic argument showing that the size of the $k-RNG$ in R^3 is $O(k^{2/3}n^{4/3})$ was given in [AM92].

Consider a subset $R \subset V$ of $r = \lfloor \frac{n}{k} \rfloor$ points; each subset chosen with equal probability. A pair of points p, q which is an edge in the $k-RNG(V)$ is also an edge in the $RNG(R)$ if $p, q \in R$ and $\Lambda_{p,q} \cap V$ is empty. The probability of such an event is at least $\binom{n-k-2}{r-2} / \binom{n}{r} = \Omega(1/k^2)$.

Therefore, the expected size of the $RNG(R)$ is bounded by $\Omega(M/k^2)$ where M is the size of the $k-RNG(V)$. However, in R^3 the size of the RNG of an $r = \lfloor n/k \rfloor$ element set is $O(r^{4/3})$, see Section 4. Hence $M = O(k^{2/3}n^{4/3})$.

Clearly, a similar argument can be used for other k-neighborhood graphs. For example a bound of $O(kn)$ can be established for the $k-RNG$ in R^2. Let us note that the same bound was also proved in [SC91b] based on the size of the k-geographic neighbors graph.

An $O(n^{5/3}\log n)$ algorithm to construct the $k-RNG$, for a fixed k, has been presented in [SC91b]. The algorithm has a traditional organization and works in two stages. At first the $k-GNG(V)$, the k - *geographical neighborhood graph* of V, is constructed. The $k-GNG$ is a generalization of geographical neighborhood graph and is defined as follows. Let W be a narrow region. A point $p \in V \cap W$ is a kth nearest geographical neighbor of v in W if there are exactly $k-1$ points of V in W which are closer to v than p. The $k-GNG$ is obtained by connecting points with their $1st$ through kth nearest geographical neighbors. The k-geographical neighborhood graph is a supergraph of the $k-RNG$ and it can have at most $18kn$ edges: see [SC91b]. The construction of the $k-GNG$ is modeled on an algorithm for the geographical neighborhood graph presented in [Yao82].

In the second stage the lunes determined by edges in the $k-GNG$ are preprocessed for supporting fast point location queries. Using the balancing technique it is possible to preprocess lunes and perform all n point locations in the lunes, in order to eliminate edges which have at least k neighbors, in time not exceeding the cost spent on constructing $k-GNG$.

Recently, Agarwal and Matoušek [AM92] showed that the $k-RNG$ of a set of n points in R^d in general position, where k is a fixed constant, can be computed in time $O(n^{2(1-\frac{1}{d+1})+\epsilon})$, $\epsilon > 0$. If points are in arbitrary position then the $k-RNG$ can be computed in time $O(n^{4/3+\epsilon})$ for points in R^2, and in time $O(n^{7/4+\epsilon})$ for points in R^3.

In Chang *et al.* [CTL90] the $k-RNG$ is studied in an interesting context of bottleneck Hamiltonian cycles. For an application of the $k-RNG$ to Euclidean bottleneck matchings see also [CTL91].

A *minmax* (*bottleneck*) Hamiltonian cycle of a graph is a Hamiltonian cycle which minimizes the length of its longest edge over all Hamiltonian cycles of this graph. Consider a complete graph induced by a set V of points in an Euclidean plane. Such a graph has a minmax Hamiltonian cycle which is contained in the $20-RNG(V)$; see [CTL90]. In fact, there is a transformation which after a finite number of steps produces, from any given minmax Hamiltonian cycle, a minmax Hamiltonian cycle which is a subgraph of the $20-RNG$. Correctness of the transformation is

based on an interesting observation related to narrow regions. Consider a set S of points contained in $cone(\vec{v}, \vec{u})$, $\angle(\vec{v}, \vec{u}) < \alpha$ and t the apex of the cone. Assume that for all points $p, q \in S$ the following properties hold, $r > 0$: $\delta(t,p) \geq r\ \delta(p,q) \geq r\ \delta(p,q) \geq \delta(t,p) - r$, and $\delta(p,q) \geq \delta(t,q) - r$. Then there exists a constant c depending only on α such that $|S| < c$; see [CTL90].

Now consider a minmax Hamiltonian cycle H and its longest edge pq not in the $k - RNG$. (The value of k can be specified; $k = 20$ in [CTL90].) Since pq is not an edge in the $k - RNG(V)$, the lune $\Lambda_{p,q}$ contains at least k points, $t_1, \ldots, t_m$, of V. If any pair t_i, t_j is connected by an edge in H then we can replace pq by a sequence pt_i, t_it_j, t_jq; the new cycle is again a minmax Hamiltonian cycle. Otherwise, let w_i be a direct predecessor of t_i in H; assume that H has a counterclockwise orientation. If k is large enough then by virtue of the above observation on the narrow regions there are two points w_i, w_j that satisfy at least one of the following conditions:

(a) $\delta(q, w_i) < \delta(p, q)$, or

(b) $\delta(w_i, w_j) < \delta(p, q)$, or

(c) $\delta(w_i, w_j) < \delta(w_i, t_i)$ or $\delta(w_i, w_j) < \delta(w_j, t_j)$.

In each case the edge pq can be replaced by other edges without increasing the length of the longest edge. For example, in case (b) the edges pq, w_it_i, w_jt_j can be substituted with pt_i, t_it_j, w_jq. Moreover, each application of this transformation replaces a longest edge not in the $k - RNG$ by some shorter edges. Hence, after a finite number of steps we obtain a minmax Hamiltonian cycle which is a subgraph of the $k - RNG$.

A related and important problem of constructing a *minmax length triangulation*, i.e., a triangulation which minimizes the longest edge, has been studied by Edelsbrunner and Tan [ET91]. They have proved that every finite point set V in R^2 has a minmax length triangulation which is a supergraph of the $RNG(V)$. The lemma suggests that construction of a minmax length triangulation can start from the $RNG(V)$. It leads immediately to a cubic time construction when existing dynamic programming algorithms for triangulations are utilized; see [Kli80]. It also provides the first polynomial time algorithm for this problem. Furthermore, as demonstrated in [ET91], an even faster quadratic time algorithm can be developed. The algorithm works for a general class of metrics.

k - Gabriel graphs: In a similar fashion *k-Gabriel graphs* can be introduced; see [SC90]. Let $\Gamma_{p,q} = B(\frac{p+q}{2}, \frac{\delta(p,q)}{2})$, i.e. $U_{p,q}$ is a diameter sphere. The edges of the k-Gabriel graph of V, the $k - GG(V)$, are defined as follows:

$$(p, q) \in E \text{ if and only if the cardinality of } \Gamma_{p,q} \cap V \text{ is less than } k.$$

Properties of these graphs and applications to bottleneck biconnected graphs, and Euclidean bottleneck matchings are discussed in [SC90] where an $O(k^2 n \log n)$ time construction for the $k - GG$ in R^2 is presented.

Constrained relative neighborhood graphs and Gabriel graphs: An interesting extension of relative neighborhood graphs has been investigated by Su and Chang [SC91a]. Let V be a set of n points in a plane and T be a set of nonintersecting line segments with their endpoints in V. Clearly, the number of segments in T is of order $O(n)$. We say that two points in V are visible if their connecting line segment does not intersect any edge in T.

The *constrained relative neighborhood graph of* $V \cup T$, $CRNG(V \cup T)$, is defined as a graph with vertices V and the set of edges E such that $(p, q) \in E$ if and only if:

1. $(p, q) \in T$, or
2. p, q are visible and $\Lambda_{p,q}$ doesn't contain points in V which are visible from both p and q.

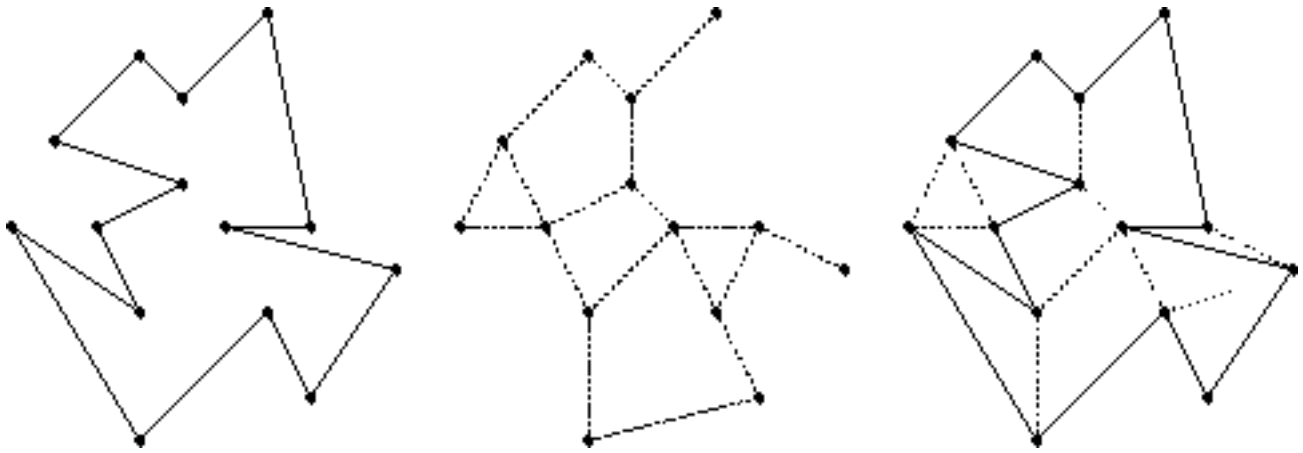

Figure 7: a polygon, the RNG of its vertices, and its $CRNG$

Figure 6 illustrates a polygon T, the $RNG(V)$ of its vertices, and its $CRNG$. Interestingly, a special case of the $CRNG$, where T forms a simple polygon, was introduced much earlier in [ET88]. Because of its applications to pattern recognition we will discuss this special case separately.

It appears that $CRNG(V \cup T)$ is a subgraph of the *constrained Delaunay triangulation* of V and T; see [SC91a]. The constrained Delaunay triangulation of V and T, $CDT(V, T)$, contains all segments in T. In addition, it includes all edges (p, q) such that p, q are visible and there is a sphere with its boundary containing p, q which does not contain points of V visible from both p and q. The $CDT(V, T)$ can be constructed in $O(n \log n)$ time; see [Aur91]. Using a process similar to Supowit's sweep it is possible to eliminate edges from $CDT(V, T) \setminus CRNG(V \cup T)$ in $O(n \log n)$ time.

Constrained Gabriel graphs, CGG, are defined in an analogous fashion. They can be obtained from the CDT in $O(n)$ time. To this end it is sufficient to consider obtuse and right angle triangles in the CDT. After eliminating from these triangles the longest edges which are not in T we obtain the desired CGG [SC91a].

Clearly, the whole concept can be extended to β-graphs. Furthermore, using the elimination forest structure of [JKY91], the constrained β-graph ($1 \leq \beta \leq 2$) can be constructed in an $O(n)$ time.

Relative neighborhood decomposition of a simple polygon: The relative neighborhood decomposition of a simple polygon is connected to constrained relative neighborhood graphs discussed earlier in this section. Let P be a simple polygon, V be its set of vertices. We set the neighborhood $U_{p,q}$ to be the lune $\Lambda_{p,q}$.

The relative neighborhood decomposition of P, $RND(P)$, is defined as a set of these edges pq which are diagonals of P and if $v \in \Lambda_{p,q}$ then at least one of pv, qv intersects the boundary of P.

The concept of the RND has been introduced by Toussaint [Tou80c]. Intuitively, we can think that an edge of P is an opaque barrier. A point can have an impact on the segment pq only if no barrier separates them.

The RND can be used to decompose simple polygons into perceptually meaningful components and is thus useful in pattern recognition.

ElGindy and Toussaint [ET88] discussed properties and several algorithms for the RND as well as the Gabriel decomposition which is defined analogously. In particular, they show that the $RND(P)$ is a planar graph and that it can be constructed in time $O(n^2)$ (n the number of edges in P.)

The RND can be viewed as a special case of a constrained relative neighborhood graph. Based on algorithms for the $CRNG$ it can be constructed in time $O(n \log n)$.

The rectangular influence graph: Ichino and Sklansky [IS85] defined the rectangular influence graph, $RIG(V)$, assuming as the neighborhood $U_{p,q}$ of points p, q the smallest coordinate oriented rectangle containing these points. They showed that the $RIG(V)$ is a supergraph of $GG(V)$ and therefore also a supergraph of the $RNG(V)$ in L_p. However, in general, the rectangular influence graph is not a subgraph of the DT.

In addition, variants such as an interclass RIG and mutual neighborhood graphs were introduced. In our terminology, the mutual neighborhood graph can be viewed as a graph defined over two point sets V_1 and V_2. The mutual graph $MNG(V_1|V_2)$ has edges for these pairs of points in V_1 which neighborhood does not contain points from V_2.

Several applications of the RIG, the interclass RIG, and the MNG are presented in [IS85]. In particular, it is argued that these graphs are useful in the design of piecewise linear classifiers, and in clustering methods applicable to mixed feature variables.

In this context we also mention direct dominance pairs which are applicable to the rectangle enclosure problem. Algorithms for reporting all direct dominance pairs are given in Gutting, Nurmi and Ottman [GNO85].

Other graphs: There is a large family of neighborhood graphs that we have not mentioned here.

This family includes such important and interesting graphs as the *sphere of influence graph* introduced by Toussaint [Tou88] and α-graphs introduced by Edelsbrunner, Kirkpatrick and Seidel [EKS83]. Both graphs are useful in *dot pattern* and *shape* analysis [Rad88]. They also have several interesting graph-theoretical properties [AH85, Tou88, EKS83]. We expand on these graphs in Section 7.

Several variants of β graphs are discussed in [KR85]. The ideas of [KR85] can be generalized and expanded on in various directions. Some discussion on possible generalizations, in particular

with an eye toward applications, is given in [Rad88].

A recent paper by Veltkamp [Vel90] proposes a parametrized family of γ-neighborhood graphs. The γ-graph can be reduced to the circle based β-skeleton, to the DT, or to the convex hull for specific parameter values.

An interesting variant of a *digital geometry* version of the RNG applied to computer vision is given in Toriwaki and Yokoi [TY88].

7 Applications

We have already touched briefly on some applications as different neighborhood graphs were introduced. In this section we consider applications in more detail. We will discuss graphs and structures which are related to the RNG as members of the same family belonging to computational morphology.

7.1 Morphology and Computer Vision

Dot Patterns The central problem in computer vision begins with a grey-level intensity array of a visual pattern or scene and culminates with a description of the scene. This description is a marriage of the intensity array itself (the bottom-up or data-driven information) and the objective for which it is viewed (the top-down or conceptually-driven information). Considerable attention has been devoted to *low-level* vision, i.e., the aspect dealing with the analysis of the intensity array up to the level of figure-background separation. The main purpose of analyzing an intensity array at this level is to form a low-level description that is independent of any final conceptually-driven description. Marr [Mar76] has called such a description a *primal sketch* of the intensity array. One class of patterns or scenes which have been studied extensively is the so-called *dot pattern*.

When dots in the plane have a finite diameter so that they are visible, and when they are fairly densely distributed in some region in the plane then a human observer is quick to perceive the "shape" of such a set. These sets are usually referred to as *dot patterns* or *dot figures*. A polygonal description of the boundary of the shape is referred to as the *shape hull* of a dot pattern, where the vertices are given as the coordinates of the dot centers.

There are two versions of the shape hull problem: in one there are no "holes" in the dot pattern and the dot pattern is "simply connected" and hence the shape hull is a simple polygon, whereas in the more difficult problem both "holes" and "disconnected" components may exist. To add to this difficulty, in some instances illusory contours are perceived between "disconnected" components as illustrated by Kennedy and Ware [KW78]. For more details and early approaches see Toussaint [Tou80a] and Medek [Med81].

In addition to describing the shape or structure of a set of points by its shape-hull or external shape we may also use the *skeleton* or *internal shape*. An early step in this direction was taken by Zahn [Zah71] with the minimal spanning tree. More recent approaches have used the RNG [IS85].

Most of the early approaches suffer from various deficiencies such as computational inefficiency, or dependence on too many parameters which must be fine-tuned in order to obtain satisfactory performance for the task at hand.

A very elegant definition for the external shape of a set of points was put forward by Edelsbrunner, Kirkpatrick, and Seidel [EKS83]. They have proposed a natural generalization of convex hulls that they call α-hulls. The α-hull of a point set is based on the notion of generalized discs in the plane. For a real number α a generalized disc of radius $\frac{1}{\alpha}$ is a (standard) disk of the radius $\frac{1}{\alpha}$ if $\alpha > 0$. If $\alpha < 0$ it is the complement of a disc of the radius $\frac{1}{\alpha}$. It is halfplane for $\alpha = 0$.

The $\alpha - hull$ of a point set S is defined to be the intersection of all closed generalized discs of radius $\frac{1}{\alpha}$ that contain all the points of S. The convex hull of S is exactly the $0 - hull$. The family

of α-hulls includes a spectrum of enclosing regions of S.

A combinatorial variant of the α-hull defined in [EKS83] is called the $\alpha-shape$. It can be viewed as the boundary of the α-hull with curved edges replaced by straight edges. Unlike the family of α-hulls, the family of distinct α-shapes has only finitely many members. They provide a spectrum of progressively more detailed descriptions of the external shape of a given point set; they can be calculated in $O(n \log n)$ time for n point sets. For more details see [EKS83] and [Ede87].

Note that the idea of α-hulls is closely related to the notions of *opening* and *closing* sets, found in mathematical morphology; see [Mat75, Ser82, Tou85]. The two dimensional α-shapes are connected to the dot patterns [Fai79, Fai83], to the circle diagrams utilized in cluster analysis [Mos67], to the muscle fiber analysis [Per88], and to shape feature analysis [STY91]. An extension of α-shapes to R^3 and $O(n^2)$ algorithm that constructs for n points a representation of the α-shapes for all values of α has been presented by Edelsbrunner and Mücke [EM92].

A new methodology for describing the *internal shape* of point sets was outlined by Kirkpatrick and Radke [KR85] where the notion of $\beta - graphs$ was introduced.

Another simple and elegant method for extracting the shape of a dot pattern is due to Rosenberg and Langridge [RL73]. The algorithm is free of parameter tuning. It is limited, however, to single objects; its computational complexity is $O(n^4)$.

Toussaint [Tou88] proposed a graph that appears to capture the essence of the *primal sketch* for dot patterns of arbitrary complexity. It seems that the graph, which is called the *sphere-of-influence* graph, suffers from none of the drawbacks of the previous methods. It delivers either the "internal" structure in the form of a "skeleton" or the "external" structure in the form of a "shape-hull' as a function of what the data look like; see Figure 7.1. Secondly, it can be applied to a scene of disconnected objects and it works without tuning of parameters. In addition, the graph affords a graph-theoretical explanation of some visual illusions such as the Mueller-Lyer illusion.

Let $V = \{p_1, \ldots, p_n\}$ be a finite set of points in the plane. For each point $p \in V$, let r_p be the closest distance to any other point in the set, and let C_p be the circle of radius r_p centered at p.

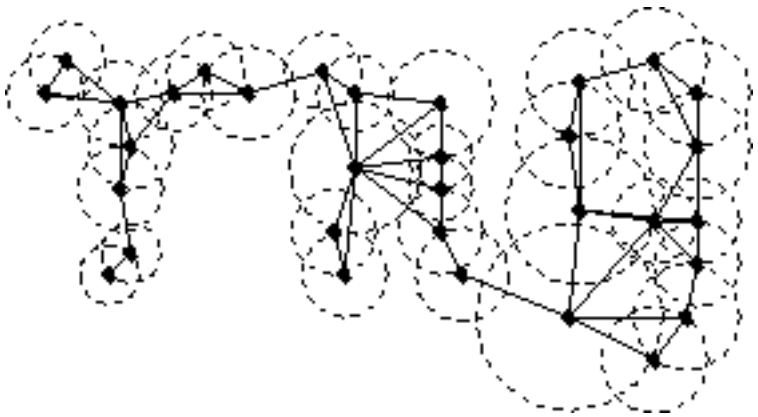

Figure 8: the sphere-of-influence graph

The sphere of influence graph is a graph on V with an edge between points p, q if and only if the circles C_p, C_q intersect in at least two places. The sphere of influence graph has at most cn edges

where c is a constant; see [AH85]. This constant is not greater than $17\frac{1}{2}$; for a discussion on this issue see [AH85, ERW89] and also [Rei48, BE51]. It implies, as was observed by ElGindy, that the algorithm of Bentley and Ottman [BO79] for reporting intersections can be used to find the sphere of influence graph in $O(n \log n)$ time. This is optimal in the decision tree model; see [AH85].

A general family of graphs related to the sphere-of-influence graph, although based on a slightly different definition and with a different purpose, has been studied in [ERW89, Rot88].

Guibas, Pach and Sharir [GPS92] have recently generalized the sphere-of-influence graph to *k-th sphere-of-influence*. Given a set V of n points in R^d, the k-th sphere-of-influence of a point $x \in V$ is the smallest closed ball centered at x and containing more than k points of V (including x). The case $k = 1$ gives the standard sphere of influence. The *k-th sphere-of-influence graph* $G_k(V)$ of V is a graph whose vertices are the points of V, and two points are connected by an edge if their k-th sphere of influence intersect. They show, extending results for 1-st sphere-of-influence graph from [AH85, ERW89], that there is a constant $c_d > 0$ depending only on d such that the number of edges in $G_k(v)$ is at most $c_d kn$. They also give an algorithm that computes the k-th sphere-of-influence graph in time $O(n^{2-\frac{2}{1+\lfloor (d+2)/2 \rfloor}+\epsilon} + kn \log^2 n)$, for any $\epsilon > 0$.

Even for $k = 1$ the sphere-of-influence graph need not be connected nor planar. In [JLM89] trees that are sphere-of-influence graphs are characterized. The geometric notion of the sphere of influence hase also been modified to a graph theoretic sense by Harary *et al.*, [HJLM90].

Texture Discrimination There are many computer vision problems where the patterns are neither dot patterns nor line pattern but *textures*. This is particularly so in the satellite image analysis of the earth.

Toriwaki and Yokoi [TY88] have applied the *RNG* and the *GG* to the problem of discriminating between different textures.

Monotonic Search Networks A graph embedded into the plane is called *monotone* if for every pair of vertices $p,\ q$ there is a path $p = v_1,\ v_2, \ldots, v_{k-1},\ v_k = q$ such that $\delta(v_i, q) > \delta(v_{i+1}, q)$ for $i = 1, \ldots, k-1$.

A useful property of monotone graphs is that the distance can be used to guide efficient traversal of the graph. Dearholt *et al.* [DGK87] have used a monotone graph called Monotonic Search Network as the underlying structure for an associative database for computer vision. Kurup [Kur91] showed that the Delaunay triangulation is a monotone graph which in general is not true for the *GG*, the *RNG* and the minimum spanning tree. He observed, however, that the *RNG* can be extended to a monotone graph, which is minimal in some sense, and consequently it can be used as a base for monotonic search networks [DGK88] which find application in the design of data bases for computer vision.

7.2 Geographic Analysis

Underlying the study of geographic analysis is *spatial analysis* [GB78] the field of study which examines the spatial structure and association of phenomena. This is a large and well established area of geography. The *RNG* and other proximity graphs may revolutionize the manner in which

the spatial analysts do their work. For a survey of applications of neighborhood graphs to spatial analysis the reader is referred to Radke [Rad88].

7.3 Pattern Classification

In the *non-parametric classification problem* a set of n feature vectors is available. It is taken from a collected data set $\{X, \Theta\} = \{(X_1, \Theta_1), \ldots, (X_n, \Theta_n)\}$, where X_i and Θ_i denote the feature vector of the *ith* object and the class label of this object, respectively. One of several powerful classification techniques is the so-called nearest-neighbor rule (NN-rule) [CH67, Dev81]. Let Y be a new object (vector feature) to be classified. The NN-rule classifies Y as belonging to class Θ_k where $X_k \in \{X_1, \ldots, X_n\}$ is the feature vector closest to Y.

In the past some practitioners avoided using the NN-rule because early algorithms were inefficient and had to store all data $\{X, \Theta\}$. Both of these problems have been eradicated with techniques from computational geometry. Various methods exist for finding nearest neighbors including an optimal $O(n \log n)$ algorithm [Vai89]. For other approaches see also [FBF77, LP77]. Furthermore, not all the "training" data $\{X, \Theta\}$ is required to be stored. Methods have been developed [TBP84] to edit redundant members of $\{X, \Theta\}$ in order to obtain a relatively small subset of $\{X, \Theta\}$ that implements exactly the same decision rule as using all of $\{X, \Theta\}$. Such methods depend heavily on the use of Voronoi diagrams and proximity graphs such as the RNG and the Gabriel graph [TBP84].

Let us finish with a few further applications of the RNG. Lee [Lee91] applied the RNG to computing shortest rectilinear paths; it is an element of a design problem in $VLSI$. In Lefkovitch [Lef87] the RNG are used in cluster analysis in ecology. Finally, in [Lef85, Lef84] the RNG found an application in comparing dissimilarity matrices.

8 Conclusion

We have reviewed in this paper results and algorithms for neighborhood graphs. Although we have focused on the relative neighborhood graphs the other members of this rich family of graphs have been discussed as well.

Many interesting questions remain open. Among them there is a problem of tight bounds for the number of edges of the RNG in R^3. The best established upper bound is superlinear. On the other hand only a trivial linear lower bound is known. In this context it would be interesting to develop optimal or output sensitive algorithms for the 3-dimensional RNG as well as the GG. Also, three dimensional β-skeletons deserve more research. Another area which needs further study is the question of recognition of proximity graphs, i.e., given a class of proximity graphs and a graph G determine if G belongs to this class. The known results concern the Delaunay triangulation [AB85] and f-factors of point sets in the plane [ERW89].

In view of the widespread applications of neighborhood graphs to computational morphology, geographical analysis, and pattern analysis the design of robust algorithms is a particularly important task. In particular, obtaining numerically stable algorithms and implementations for the neighborhood graphs is a challenging problem.

Acknowledgement We would like to thank the anonymous referees whose valuable remarks were essential in improving the presentation of our paper. The first author would like also to thank Kate Senn for her professional help in locating many of the references.

References

[AB85] P. F. Ash and E. D. Bolker. Recognizing Dirichlet tessellations. *Geometria Dedicata*, 19:175–206, 1985.

[AESW90] P. Agarwal, H. Edelsbrunner, O. Schwarzkopf, and E. Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. In *Proc. of the 6th ACM Symp. on Computational Gometry*, pages 203–210, 1990.

[AM92] P. K. Agarwal and J. Matoušek. Relative neighborhood graphs in three dimensions. In *Proc. 3rd Annual ACM-SIAM Symp. Discrete Algorithms*, pages 58–67, 1992. (Accepted to *Computational Geometry: Theory and Applications*).

[AGSS87] A. Aggarwal, L.J. Guibas, J. Saxe, and P.W. Shor. A linear time algorithm for computing the Voronoi diagram of a convex polygon. In *Proc. of the 19th Annual ACM Symposium on Theory of Computing*, pages 39–45, 1987.

[AH85] D. Avis and J. Horton. Remarks on the sphere of influence graph. In J. E. Goodman et al., editor, *Discrete Geometry and Convexity*, pages 323–327. New York Academy of Sciences, 1985.

[Aur88] F. Aurenhammer. Improved algorithms for discs and balls using power diagrams. *J. Algorithms*, 9:151–161, 1988.

[Aur91] F. Aurenhammer. Voronoi diagrams - a survey of fundamental geometric data structure. *ACM Computing Surveys*, 23:345–406, 1991.

[BE51] P. Bateman and P. Erdös. Geometrical extrema suggested by a lemma of Besicovitch. *Amer. Math. Monthly*, 58:306–314, 1951.

[BO79] J. Bentley and T. Ottman. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Computers*, C-28:643–647, 1979.

[BWY80] J. L. Bentley, B. W. Weide, and A. C. Yao. Optimal expected-time algorithms for closest point problems. *ACM Trans. Math. Software*, pages 563–580, 1980.

[BBD87] J. W. Boland, R.C. Brigham, and R. D. Dutton. The difference between a neighborhood graph and a wheel. *Congressus Numerantium*, 58:151–156, 1987.

[CTL90] M. S. Chang, C.Y. Tang, and R.C.T. Lee. 20-relative neighborhood graphs are Hamiltonian. In *Proceedings of the International Symposium SIGAL '90*, pages 53–65, August 1990.

[CTL91] M. Chang, C. Tang, and R. Lee. Solving the Euclidean bottleneck matching problem by k-relative neighborhood graph. Accepted to *Algorithmica*, 1991.

[CEG+90] B. Chazelle, H. Edelsbrunner, L.J. Guibas, J.E. Hershberger, R. Seidel, and M. Sharir. Slimming down by adding; selecting heavily covered points. In *Proc. of 6th Annual Symposium on Computational Geometry*, pages 116–127. 1990.

[Cha86] B. Chazelle. Filtering search: A new approach to query answering. *SIAM J. Computing*, 15:703–724, 1986.

[CSW90a] B. Chazelle, M. Sharir, and E. Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorem. In *Proc. of the 6th ACM Symp. on Computational Gometry*, pages 23–33, 1990.

[CEG+89] K.L. Clarkson, H. Edelsbrunner, L.J. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and surfaces. In *Proc. of the 29th Annual Symposium on Foundations of Computer Science*, pages 568–579. 1989.

[CSW90b] K. L. Clarkson, H. Edelsbrunner L. J. Guibas M. Sharir, and E. Welzl. Combinatorial complexity bounds and arrangements of curves and surfaces. *Discrete and Computational Geometry*, 5:99–160, 1990.

[Chu89] F. R. K. Chung. Sphere-and-point incidence relations in high dimensions with applications to unit distances and furthest-neighbor pairs. *Discrete and Computational Geometry*, 4:183–190, 1989.

[CH67] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Information Theory*, IT-13:21–27, 1967.

[Dev81] L. P. Devroye. On the inequality of Cover and Hart in nearest neighbor discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3:75–78, 1981.

[Dev88] L. P. Devroye. The expected size of some graphs in computational geometry. *Computers and Mathematics with Applications*, 15:53–64, 1988.

[DGK87] D. W. Dearholt, N. Gonzales, and H. Kirpekar. Associative network databases for computer vision. In *Proc. 21st Asilomar Conference on Signals, Systems, and Computers*, pages 859–864, 1987.

[DGK88] D. W. Dearholt, N. Gonzales, and G. Kurup. Monotonic search networks for computer vision databases. In *Proc. 22nd Asilomar Conference on Signals, Systems, and Computers*, pages 548–553, 1988.

[DH91] D. W. Dearholt and F. Harary, editors. *Proceedings of the First Workshop on Proximity Graphs*. Computing Research Laboratory, New Mexico State University, 1991. Memoranda in Computer and Cognitive Science MCCS-91-224.

[Dod72] C. W. Dodge. *Euclidean geometry and transformations.* Addison-Wesley, 1972.

[Dun90] W. Dunham. *Journey Through Genius: The Great Theorems of Mathematics.* John Wiley and Sons, Inc., 1990.

[Dwy88] R. Dwyer. *Average-case analysis of algorithms for convex hulls and Voronoi diagrams.* PhD thesis, Carnegie-Mellon University, 1988.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, 1987.

[EKS83] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. Inform. Theory*, IT-29:551–559, 1983.

[EM92] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. Manuscript, 1992.

[EO85] H. Edelsbrunner and M. H. Overmars. Batched dynamic solutions to decomposable searching problems. *J. Algorithms*, 6:515–542, 1985.

[ERW89] H. Edelsbrunner, G. Rote, and E. Welzl. Testing the necklace condition for shortest tours and optimal factors in the plane. *Theoretical Computer Science*, 66:157–180, 1989.

[ES90] H. Edelsbrunner and M. Sharir. A hyperplane incidence problem with applications to counting distances. In *Proc. of International Symposium on Algorithms, LNCS 450*, pages 419–428. Springer-Verlag, 1990.

[ET91] H. Edelsbrunner and T. S. Tan. A quadratic time algorithm for the MinMax length triangulation. Technical report, Dept. of Comp. Sci. at University of Illinois at Urbana Champaign, February 1991.

[ET88] H.A. ElGindy and G.T. Toussaint. Computing the relative neighborhood decomposition of a simple polygon. *Computational Morphology*, pages 53–70, 1988.

[Fai79] J. Fairfield. Contoured shape generation: forms that people see in dot patterns. In *Proc of the IEEE conference on Cybernetics and Society*, pages 60–64, 1979.

[Fai83] J. Fairfield. Segmenting dot patterns by voronoi diagram concavity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 104–110, 1983.

[FBF77] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Software*, 3:209–226, 1977.

[GB78] A. Getis and B. N. Boots. *Models and Spatial Process.* Cambridge University Press, Cambridge, 1978.

[GBT84] H. N. Gabow, J. L. Bentley, and R. E. Tarjan. Scaling and related techniques for geometry problems. In *Proc. 16th Annual ACM Symposium on Theory of Computing*, pages 135–143, 1984.

[GNO85] R. H. Gutting, O. Nurmi, and T. Ottmann. The direct dominance problem. In *Proc. of the First Annual ACM Symposium on Computational Geometry*, pages 81–88, 1985.

[GPS92] L. Guibas, J. Pach, and M. Sharir. Generalized sphere-of-influence graphs in higher dimensions. Manuscript, Tel-Aviv University, 1992.

[GS69] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[GS83] L. J. Guibas and J. Stolfi. On computing all North-East nearest neighbors in the l_1 metric. *Info. Proc. Letters*, 17:219–223, 1983.

[GT85] H. N. Gabow and R. E. Tarjan. A linear-time algorithm for a special case of disjoint set union. *J. of Computer and System Sciences*, 30:209–221, 1985.

[HJLM90] F. Harary, M. S. Jacobson, M. J. Lipman, and F. R. McMorris. Sphere-of-influence graphs defined on prescribed graph. Manuscript, University of Louisville, 1990.

[Hwa90] N.-F. Hwang. Divide and conquer algorithm for RNG. *BIT*, 30:196–206, 1990.

[IIM85] H. Imai, M. Iri, and K. Murota. Voronoi diagram in the Laguerre geometry and its applications. *SIAM J. Comp.*, 14:93–105, 1985.

[IS85] M. Ichino and J. Sklansky. The relative neighborhood graph for mixed feature variables. *Pattern Recognition*, 18:161–167, 1985.

[JK87] J. W. Jaromczyk and M. Kowaluk. A note on relative neighborhood graphs. In *Proceedings 3rd Annual Symposium on Computational Geometry*, pages 233–241, Waterloo, Canada, June 8-10, 1987. ACM Press.

[JK91] J.W. Jaromczyk and M. Kowaluk. Constructing the relative neighborhood graph in 3-dimensional Euclidean space. *Discrete Applied Math.*, 31:181–192, 1991.

[JKY91] J.W. Jaromczyk, M. Kowaluk, and F. Yao. An optimal algorithm for constructing β-skeletons in l_p metric. Accepted to *SIAM J. Comput.*, 1991.

[JLM89] M. S. Jacobson, M. J. Lipman, and F. R. McMorris. Trees that are sphere-of-influence graphs. Technical report, University of Louisville, 1989.

[Kat87] J. Katajajnen. *Bucketing and Filtering in Computational Geometry.* PhD thesis, University of Turku, Finland, 1987. (Available as Report A46).

[Kat88] J. Katajainen. The region approach for computing relative neighbourhood graphs in the l_p metric. *Computing*, 40:147–161, 1988.

[Kli80] G. T. Klinscek. Minimal triangulations of polygonal domains. *Annals Discrete Math.*, pages 121–123, 1980.

[KN86] J. Katajainen and O. Nevalainen. Computing relative neighbourhood graphs in the plane. *Pattern Recognition*, 19:221–228, 1986.

[KN87] J. Katajainen and O. Nevalainen. An almost naive algorithm for finding relative neighbourhood graphs in l_p metrics. *Informatique Theoretique et Applications*, 21:199–215, 1987.

[KNT87] J. Katajainen, O. Nevalainen, and J. Teuhola. A linear expected-time algorithm for computing planar relative neighbourhood graphs. *Information Processing Letters*, 25:77–86, 1987.

[KR85] D. G. Kirkpatrick and J. D. Radke. A framework for computational morphology. In G. Toussaint, editor, *Computational Geometry*, pages 217–248. North-Holland, 1985.

[Kur91] G. Kurup. *A Database Organized on the Basis of Similarities with Applications in Computer Vision.* PhD thesis, New Mexico State University, June 1991.

[KW78] J. M. Kennedy and C. Ware. Illusory contours can arise in dot figures. *Perception*, pages 191–194, 1978.

[Lan69] P. M. Lankford. Regionalization: theory and alternative algorithms. *Geogr. Annals*, 1:196–212, 1969.

[Law82] R. Lawror. *Sacred Geometry: Philosophy and Practice.* Thames and Hudson Ltd., London, 1982.

[Lee80] D. T. Lee. Two dimensional voronoi diagram in the l_p metric. *J. of ACM*, 27:604–618, 1980.

[Lee85] D.T. Lee. Relative neighborhood graphs in the l_1-metric. *Pattern Recognition*, 18:327–332, 1985.

[Lee91] D.T. Lee. Rectangle proximity graphs and rectilinear shortest path problems. In D. W. Dearholt and F. Harary, editors, *Proceedings of the First Workshop on Proximity Graphs. Memoranda in Computer and Cognitive Science MCCS-91-224*, pages 57–87. Computing Research Laboratory, New Mexico State University, Las Cruces, 1991.

[Lef84] L. P. Lefkovitch. A nonparametric method for comparing dissimilarity matrices, a general measure of biogeographical distance, and their applications. *Amer. Natur.*, 123:484–499, 1984.

[Lef85] L. P. Lefkovitch. Further nonparametric tests for comparing dissimilarity matrices based on the relative neighborhood graph. *Mathematical Biosciences*, 73:71–88, 1985.

[Lef87] L. P. Lefkovitch. Species associations and conditional clustering; clustering with or without pairwise resemblances. In P. Legendre and L. Legendre, editors, *NATO ASI Series: Developments in Numerical Ecology*, volume G14, pages 309–331. Springer-Verlag, 1987.

[LP77] D. T. Lee and F. P. Preparata. Location of a point in a planar subdivision and its applications. *SIAM Journal of Computing*, pages 594–606, 1977.

[Mar76] D. Marr. Early processing for of visual information. *Philosophical Transactions of the Royal Socia ety of London*, 275(945):483–524, 1976.

[Mat75] G. Matheron. *Random Sets and Integral Geometry*. John Wiley, 1975.

[Mau84] A. Maus. Delaunay triangulation and the convex hull of points in expected linear time. *BIT*, 24:151–163, 1984.

[Med81] V. Medek. On the boundary of a finite set of points in the plane. *Computer Graphics and Image Processing*, 15:93–99, 1981.

[Mil70] R. E. Miles. On the homogenous planar poisson point process. *Mathematical Bioscences*, 6:85–127, 1970.

[Mos67] W. W. Moss. Some new analytic and graphic approaches to numerical taxonomy, with an example from dermanyssidae (acari). *Systematic Zoology*, 16:177–207, 1967.

[MS84] D. W. Matula and R. R. Sokal. Properties of Gabriel graphs relevant to geographical variation research and the clustering of points in the plane. *Geographical Analysis*, 12:205–222, 1984.

[Ola89] S. Olariu. A simple linear-time algorithm for computing the *rng* and *mst* of unimodal polygons. *Information Processing Letter*, 31:243–248, 1989.

[O'R82] J. O'Rourke. Computing the relative neighborhood graph in the l_1 and l_∞ metrics. *Pattern Recognition*, pages 45–55, 1982.

[OW88] M. H. Overmars and E. Welzl. New methods for computing visibility graphs. In *Proc. the Fourth Annual Symposium on Comp. Geometry*, pages 164–171, 1988.

[Per88] F. Pernus. The Delaunay triangulation and the shape hull as tools in muscle fiber analysis. *Pattern Recognition Letters*, 8:197–202, 1988.

[PS86] F. Preparata and M. I. Shamos. *Computational Geometry – Introduction*. Springer-Verlag, New York, 1986.

[Rad88] J. D. Radke. On the shape of a set of points. In G. T. Toussaint, editor, *Computational Morphology*, pages 105–136. North-Holland, 1988.

[Rei48] E. R. Reinferberg. A problem on circles. *Math. Gaz.*, 32:290–292, 1948.

[RL73] B. Rosenberg and D. J. Langridge. A computational view of perception. *Perception*, pages 415–424, 1973.

[Rot88] G. Rote. *Two solvable cases of the Traveling Salesman Problem.* PhD thesis, Technische Universitat Graz, 1988.

[STY91] T. Saito, J.-I. Toriwaki, and S. Yokoi. Properties of extended digital α-hull with applications to shape feature analysis of a figure set. *Forma*, 6:9–25, 1991.

[Ser82] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.

[Smi89] W. D. Smith. *Studies in computational geometry motivated by mesh generation.* PhD thesis, Princeton University, 1989.

[SC90] T.-H. Su and R.-Ch. Chang. The k-Gabriel graphs and their applications. In *Proceedings of the International Symposium SIGAL '90*, pages 66–75, August 1990.

[SC91a] T.-H. Su and R.-Ch. Chang. Computing the constrained relative neighborhood graphs and constrained Gabriel graphs in Euclidean plane. *Pattern Recognition*, 24:221–230, 1991.

[SC91b] T.-H. Su and R.-Ch. Chang. Computing the k-relative neighborhood graphs in Euclidean plane. *Pattern Recognition*, 24:231–239, 1991.

[SC91c] T.-H. Su and R.-Ch. Chang. On constructing relative neighborhood graphs in Euclidean k-dimensional spaces. *Computing*, 46:121–130, 1991.

[Sup83] K. J. Supowit. The relative neighborhood graph, with an application to minimum spanning trees. *J.Assoc.Comput.Mach.*, 30:428–448, 1983.

[Tar83] R. E. Tarjan. *Data Structures and Network Algorithms.* SIAM, Philadelphia, 1983.

[TBP84] G. T. Toussaint, B. K. Bhattacharya, and R. S. Poulsen. The application of Voronoi diagrams to non-parametric decision rules. In *Proc. Computer Science and Statistics: 16 Symposium on the Interface*, Atlanta, March 14-16 1984.

[TM80] G. T. Toussaint and R. Menard. Fast algorithms for computing the planar relative neighborhood graph. In *Proc. 5th Symposium on Operations Research*, pages 425–428, 1980.

[Tou80a] G. T. Toussaint. Pattern recognition and geometric complexity. In *Proc. 5th International Conference on Pattern Recognition*, pages 1324–1347, Miami Beach, December 1980.

[Tou80b] G. T. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.

[Tou80c] G. T. Toussaint. Decompossing a simple polygon with the relative neighbourhood graph. In *Proceedings of the Allerton Conference*, pages 20–28, October 1980.

[Tou85] G. T. Toussaint. Computational geometry and morphology. In *Proc. First International Symposium for Science and Form*, pages 395–403, Tsukuba, Japan, November 1985.

[Tou88] G.T. Toussaint. A graph-theoretical primal sketch. In G. T. Toussaint, editor, *Computational Morphology*, pages 229–260. North-Holland, 1988.

[Tou91] G. T. Toussaint. Some unsolved problems on proximity graphs. In D. W. Dearholt and F. Harary, editors, *Proceedings of the First Workshop on Proximity Graphs. Memoranda in Computer and Cognitive Science MCCS-91-224*. Computing research laboratory, New Mexico State University, Las Cruces, 1991.

[TY88] J. I. Toriwaki and S . Yokoi. Voronoi and related neighbors on digitized two dimensional space with application to texture analysis. In G. T. Toussaint, editor, *Computational Morphology*, pages 207–228. North-Holland, 1988.

[Urq80] R. B. Urquhart. Algorithms for computation of relative neighborhood graph. *Electronic Letters*, 14(14):556–557, 1980.

[Urq83] R. B. Urquhart. Some properties of the planar Euclidean relative neighborhood graph. *Pattern Recognition Letters*, pages 317–322, 1983.

[Vai89] P. M. Vaidya. An $O(n \log n)$ algorithm for the all-nearest-neighbors problem. *Discrete and Computational Geometry*, pages 101–115, 1989.

[Vel90] R. C. Veltkamp. The γ-neighborhood graph. Technical report, Centre for Mathematics and Computer Science, July 1990.

[Wee89] Y. C. Wee. On the angle restricted nearest neighbor problem. Technical Report 89-8, Dept. Comp. Sci., University at Albany, 1989.

[Wil85] D. E. Willard. New data structures for orthogonal range queries. *SIAM J. Comput.*, 14:232–253, 1985.

[Yao82] A. C.-C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Comput.*, 11:721–736, 1982.

[Zah71] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Computers*, C-20:68–86, 1971.