Computing 40, 147 – 161 (1988)

Computing

The Region Approach for Computing Relative Neighbourhood Graphs in the L_p Metric

J. Katajainen, Turku

Received February 18, 1987; revised September 22, 1987

Abstract — Zusammenfassung

The Region Approach for Computing Relative Neighbourhood Graphs in the L_p Metric. The following geometrical proximity concepts are discussed: relative closeness and geographic closeness. Consider a set $V = \{v_1, v_2, \ldots, v_n\}$ of distinct points in a *two-dimensional space*. The point v_j is said to be a *relative neighbour* of v_i if

$$d_p(v_i, v_j) \leq \max\{d_p(v_i, v_k), d_p(v_j, v_k)\} \quad \text{for all } v_k \in V,$$

where d_p denotes the distance in the L_p metric, $1 \leq p \leq \infty$. After dividing the space around the point v_i into eight sectors (regions) of equal size, a closest point to v_i in some region is called an *octant* (*region*, or *geographic*) *neighbour* of v_i. For any L_p metric, a relative neighbour of v_i is always an octant neighbour in some region at v_i. This gives a direct method for computing all relative neighbours, i.e. for establishing the *relative neighbourhood graph* of V. For every point v_i of V, first search for the octant neighbours of v_i in each region, and then for each octant neighbour v_j found check whether the point v_j is also a relative neighbour of v_i. In the L_p metric, $1 < p < \infty$, the total number of octant neighbours is shown to be $\theta(n)$ for any set of n points; hence, even a straightforward implementation of the above method runs in $\theta(n^2)$ time. In the L_1 and L_∞ metrics the method can be refined to a $\theta(n \log n + m)$ algorithm, where m is the number of relative neighbours in the output, $n - 1 \leq m \leq n(n-1)$. The L_1 (L_∞) algorithm is optimal within a constant factor.

Key words: Region approach, relative neighbourhood graphs, range searching, computational geometry, analysis of algorithms.

Bereiche zur Berechnung von relativen Nachbarschaftsgraphen in der L_p-Metrik. Folgende Konzepte für geometrische Nähe werden diskutiert: relative Nähe und geographische Nähe. Sei $V = \{v_1, \ldots, v_n\}$ eine Menge von Punkten eines zweidimensionalen Raumes. Ein Punkt v_j heißt relativer Nachbar von v_i, falls

$$d_p(v_i, v_j) \leq \max\{d_p(v_i, v_k), d_p(v_j, v_k)\} \quad \text{für alle } v_k \in V,$$

wobei d_p den Abstand in der L_p-Metrik bezeichnet, $1 \leq p \leq \infty$. Nach Unterteilung des Raumes um v_i in acht Sektoren (Bereiche) gleicher Größe heißt ein nächster Punkt von v_i in einem Bereich ein Oktant- (Bereichs- oder geographischer) Nachbar von v_i. Für jede L_p-Metrik ist ein relativer Nachbar von v_i auch immer ein Oktant-Nachbar in irgendeinem Bereich um v_i. Dies liefert eine direkte Methode zur Berechnung aller relativen Nachbarn, d. h. zur Erstellung der relativen Nachbarschaftsgraphen von V. Für jeden Punkt v_i von V werden zuerst die Oktant-Nachbarn von v_i in jedem Bereich gesucht, dann wird für jeden gefundenen Oktant-Nachbarn v_j geprüft, ob v_j auch ein relativer Nachbar von v_i ist. Er wird gezeigt, daß in der L_p-Metrik, $1 < p < \infty$, die Gesamtanzahl der Oktant-Nachbarn $\theta(n)$ ist für jede Menge mit n Punkten; daher läuft auch eine nicht optimierte Implementierung der oben beschriebenen Methode in $\theta(n^2)$ Zeit. In der L_1 und L_∞ Metrik kann die Methode zu einem $\theta(n \log n + m)$ Algorithmus verfeinert werden, wobei m die Anzahl der relativen Nachbarn des Ergebnisses ist, $n - 1 \leq m \leq n(n-1)$. Der L_1 (L_∞)-Algorithmus ist optimal bis auf einen konstanten Faktor.

1. Introduction

Let V be a set of n distinct points in a metric space (X, d), where X is a non-empty set and d a distance measure. Two points v and w of V are said to be *relative neighbours* if, and only if,

$$d(v, w) \leq \max\{d(u, v), d(u, w)\} \quad \text{for all } u \in V.$$

The definition states that two points are relative neighbours if they are at least as close to each other as they are to any other point. The geometrical interpretation for this is that the region

$$\text{lune}(v, w) = \{x \in X \mid \max\{d(x, v), d(x, w)\} < d(v, w)\},$$

which is the intersection of two open balls, both of radius equal to $d(v, w)$ centered at v and w, contains no points of V.

The *relative neighbourhood graph* (RNG) of V connects all relative neighbours, i.e. two points are joined by an edge if, and only if, they are relative neighbours. The RNG was introduced by Toussaint [20] as a possible tool for pattern recognition and data analysis problems. For the applications of RNG the reader is referred to [19, 21, 22, 24, 28].

The present is a study of the problem of computing the relative neighbourhood graph for a given set of points in $\Re_p^2$, the two-dimensional space with the L_p metric. Almost exclusively the RNG-algorithms proposed in the literature follow the same pattern: first a supergraph of RNG is computed, and then the supergraph is reduced to RNG. Table 1 summarizes the worst-case performance characteristics of some RNG-algorithms running in the plane. The average case performance of some RNG-algorithms has been discussed in [11, 12, 13, 25]. The multidimensional RNG-algorithms, which are omitted here, have been studied in [9, 12, 16, 19, 26 (see also [23])].

Table 1. *Characteristics of different RNG-algorithms in the plane*

Metric	Supergraph	Reduction method	Complexity	Reference
L_2	Voronoi dual	Brute force	$\theta(n^2)$	[20]
	Voronoi dual	Scan line approach	$\theta(n \log n)$	[19]
	Region neighbour graph	Brute force	$\theta(n^2)$	[11]
L_p $1<p<\infty$	Complete graph	Brute force	$\theta(n^3)$	[20]
	Near neighbour graph	Brute force	$O(n^{2.5})$; $\theta(n^2)$*	[9, 12]
	Delaunay triangulation	Elimination forest of edges	$\theta(n \log n)$**	[9, 14]
	Region neighbour graph	Brute force	$\theta(n^2)$	this paper
L_1 and L_∞	Complete graph	Range searching	$\theta(n^2 \log n)$	[16]
	Delaunay triangulation	Scan line approach	$\theta(n \log n)$**	[15]
	Region neighbour graph	Range searching	$\theta(n \log n + m)$	this paper

* When no three points form an isosceles triangle.
** When no four points are cocircular.

Several closest point problems can be solved efficiently by the region approach (for the terminology, see [7]); for example, it can be used for computing minimum spanning trees [7, 8, 30], relative neighbourhood graphs [11, 13, 19], or Voronoi diagrams [3]. The method is to divide the space around each input point into a finite set of narrow regions (e.g. in the plane into eight sectors of equal size) such that the number of regions is dependent only on the metric space, but independent of the cardinality of the given point set. After finding the closest points, called the *region neighbours*, in each region for all points, the problems can be solved without considering (many) other connections to points further away. To construct minimum spanning trees it is enough to find only *one* of the region neighbours in each region, and ties can be resolved in an arbitrary fashion. However, in order to compute relative neighbourhood graphs correctly it is necessary to find *all* the region neighbours in each region.

The region approach was proposed by Supowit [19] as a possible way of computing RNG in a multidimensional Euclidean space. He proved that, with a special partition of the space, the graph connecting each point to all of its region neighbours is a supergraph of RNG. Now the supergraph can be reduced to RNG by using the brute force reduction, where for each edge of the supergraph it is checked whether there is a point of V which falls inside the lune. Supowit showed that this simple method runs in $\theta(n^2)$ time on the assumption that for all points in each narrow region, the region neighbour is unique. The method has been further studied in the Euclidean plane by Katajainen and Nevalainen [11]. The total number of region neighbours was proved to be at most $\theta(n)$ for any point set, and this leads immediately to a $\theta(n^2)$ RNG-algorithm without the uniqueness assumption.

In the present paper it is shown that the planar RNG-algorithm of Katajainen and Nevalainen [11] will work in $\theta(n^2)$ time for any L_p metric, $1 < p < \infty$. Also a refined algorithm is introduced, which runs in the L_1 and L_∞ metrics, and is of time complexity $\theta(n \log n + m)$, where m denotes the number of edges in the resulting RNG. Observe that in the L_p metric, $1 < p < \infty$, RNG has at most $\theta(n)$ edges, but in the L_1 and L_∞ metrics there exist point sets for which the RNG may have $\Omega(n^2)$ edges, i.e. $n-1 \leq m \leq n(n-1)$. The L_∞ (L_1) algorithm combines an efficient method for finding an L_∞ region neighbour (see, e.g. [7, 8]) with the reduction method, using range searching as proposed by O'Rourke [16]. That is to say, in the L_∞ metric the lune is a rectangle and thus the lune test is a special range query for which efficient data structures are known [1, 2, 4, 7, 29]. The time and space complexities of the L_∞ RNG-algorithm are optimal when the constant factors are not taken into account.

The paper is organised as follows. Section 2 summarizes what is known about the close connection between the relative neighbours and region neighbours. Section 3 explores the partition of the plane into eight sectors of equal size, and the octant neighbour graph induced by the partition. Section 4 describes and analyses the RNG-algorithms based on the above partition. Section 5 contains some concluding remarks.

The model of computation used throughout this paper is the standard real RAM, i.e. a random access machine with infinite precision, real number, arithmetic (for details, see [18]).

2. Region Neighbours Versus Relative Neighbours

This section draws attention to the close relationship between relative neighbours and region neighbours.

Consider a metric space (X, d), where X is a non-empty set and d a distance measure. Let us assume that, for any point $v \in X$, the space around v can be covered completely by a finite set of narrow regions. For a point v, region $R(v)$ is said to be *narrow* if for any two points $x, y \in R(v)$,

$$d(x, y) < \max\{d(v, x), d(v, y)\}.$$

Let V be a set of points in X. If $v \in V$, a closest point of $V \setminus \{v\}$ in some narrow region at v, is a *region neighbour* of v.

The correctness of our algorithms is based on the following:

Theorem 1: *Let V be a set of points in the metric space (X, d). If a collection $\rho(v)$ of narrow regions at a point $v \in V$ covers the space completely, then a relative neighbour of v is a region neighbour of v in some region $R(v)$ of $\rho(v)$.*

Proof: The argument follows directly from the definitions for relative neighbours and narrow regions. Let u be a relative neighbour of v and assume that the point u is located in a region $R(v)$ with respect to v. If w is another point of V in the region $R(v)$, we have

$$d(u, v) \leq \max\{d(u, w), d(v, w)\} = d(v, w).$$

This is because the points u and v are relative neighbours, and the region $R(v)$ is narrow. Hence, no point of V can be closer to v than the point u in $R(v)$. This shows that u is a region neighbour, but not necessarily a unique one. □

Although the above result holds true very generally, we apply it only to Cartesian coordinate spaces. Yao [30] proved that for any point v in a multidimensional coordinate space with the L_p metric, the space around v can be covered by a finite set of narrow, convex polyhedral cones, centered at v. Gabow, Bentley and Tarjan [7] improved Yao's result by presenting several sparser partitions that can be explicitly stated.

Theorem 1 has practical significance because in some cases, instead of finding all relative neighbours directly, it might be easier first to search for all region neighbours and then exclude those which are not relative neighbours. In order for the method to be efficient, the total number of narrow regions and the total number of all region neighbours must not be too large. In the two-dimensional space this is indeed the case, and the region-based approach is therefore efficient, as will be seen in the following sections.

3. All Octant Neighbours

In this section we explore the partition of a two-dimensional space into eight sectors of equal size, and the octant neighbour graph induced by that partition. First, it is shown that the partition divides the space into narrow regions. Second, the total number of all octant neighbours is shown to be $\theta(n^2)$ in the L_1 and L_∞ metrics and $\theta(n)$ in the L_p metric, $1 < p < \infty$.

3.1. Narrow Regions in the Plane

In $\mathfrak{R}_p^2$ the distance between two points $v=(v_x, v_y)$ and $w=(w_x, w_y)$ is

$$d_p(v,w)=(|v_x-w_x|^p+|v_y-w_y|^p)^{1/p}, \text{ for } 1\leq p<\infty;$$

and

$$d_\infty(v,w)=\max\{|v_x-w_x|, |v_y-w_y|\}.$$

Let S be a region in $\mathfrak{R}_p^2$ and let $\mathrm{rot}(\alpha, S)$ denote the set obtained by rotating the points of S α degrees counterclockwise around v. Consider the collection $\rho_p^2(v)$ of regions,

$$\rho_p^2(v)=\{R_1(v), R_2(v), \ldots, R_8(v)\},$$

which divides the space around v into eight distinct regions of equal size as follows (see Fig. 1 a):

$$R_1(v)=\{u\in\mathfrak{R}_p^2 \mid u_x\geq v_x, u_y>v_y, u_x-v_x>u_y-v_y\};$$

$$R_i(v)=\mathrm{rot}(45°, R_{i-1}(v)), \ i=2,3,\ldots,8.$$

Observe that R_1 is closed on the x-axis but open on the diagonal.

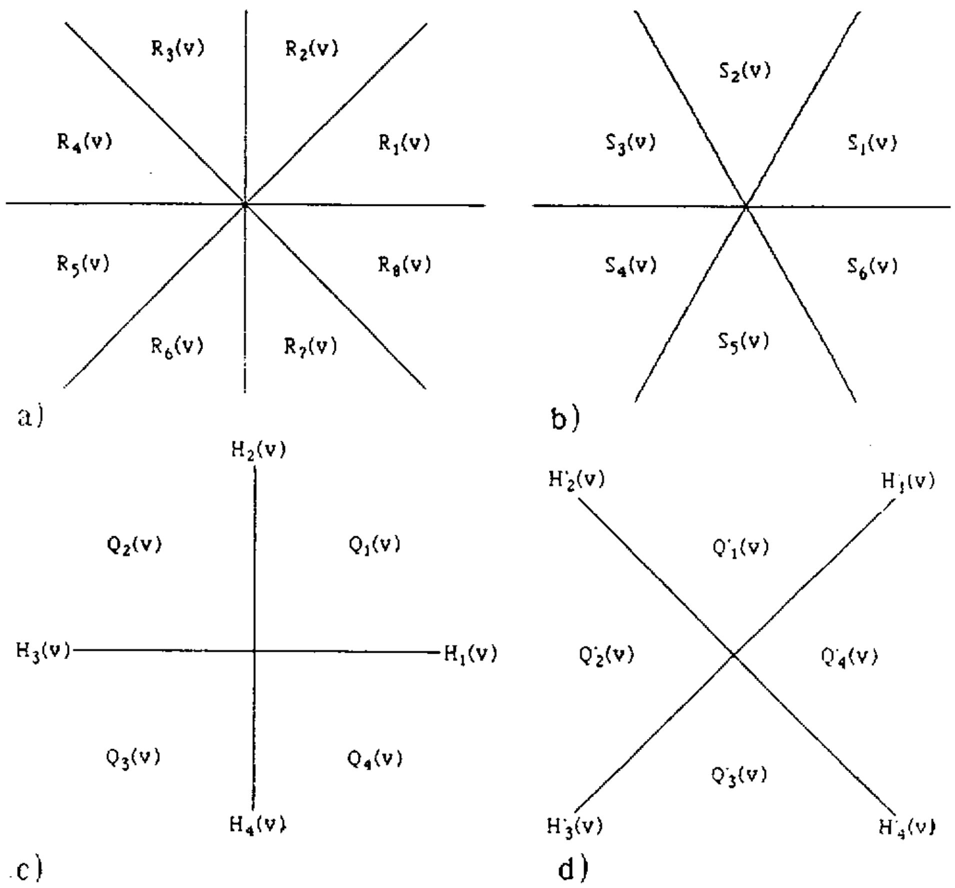

Fig. 1. Some narrow partitions a) $\rho_p^2(v)$, b) $\rho_2^2(v)$, c) $\rho_\infty^2(v)$, and d) $\rho_1^2(v)$

It should be noted that in the literature some other partitions into narrow regions have been proposed. These could equally well be used. For example, the partition $\rho_2^2(v)$, which divides the plane into six sectors of equal size (Fig. 1 b) [11]; the partition $\rho_\infty^2(v)$, which divides the plane into eight regions, four quadrants and four half-lines (Fig. 1 c) [7]; or the partition $\rho_1^2(v)$, which is obtained by rotating the $\rho_\infty^2(v)$ regions 45° degree counterclockwise (Fig. 1 d) [7]. Each of the partitions $\rho_2^2(v)$, $\rho_\infty^2(v)$, and $\rho_1^2(v)$ divide the plane into narrow regions for L_2, L_∞, and L_1 metrics, respectively. By studying the partition $\rho_\infty^2(v)$ (or $\rho_1^2(v)$) more closely, the number of narrow regions can easily be reduced from eight to five in $\Re_\infty^2$ ($\Re_1^2$).

In spite of the fact that partitions with a lower number of regions exist, there are two reasons why we consider only the partition $\rho_p^2(v)$ hereafter:

1. The regions of $\rho_p^2(v)$ are narrow for any L_p metric, $1 \le p \le \infty$, as will be shown in the next theorem.
2. In the algorithms based on the region approach, we frequently have to determine the region in which a given test point lies with respect to a point v. For the partition $\rho_p^2(v)$ this operation demands at most three subtractions, one of which is unary, and three comparisons. The verification of this fact is left to the reader.

Theorem 2: *All regions of the partition $\rho_p^2(v)$ are narrow in $\Re_p^2$, $1 \le p \le \infty$.*

Proof: Essentially the technique was given in [7, Region Theorem] where the general partition, if applied to the two-dimensional case, divides the plane into 16 narrow regions, 8 octants and 8 half-lines. For convenience, let us assume that the point v is the origin of $\Re_p^2$. Because of the symmetry, it is sufficient to prove the argument for the first octant $R_1(v)$. Further let us assume that $1 \le p < \infty$. The case $p = \infty$ follows from the isometry of the two-dimensional spaces $\Re_1^2$ and $\Re_\infty^2$ [5].

Let u and w be two points in $R_1(v)$. We can rename the points u and w such that $u_x \ge w_x$. If we can show that

$$d_p(u, w) < \max\{d_p(v, u), d_p(v, w)\}$$

then the result follows. Now two cases arise.

Case 1: $u_y \ge w_y$.
Then

$$\begin{aligned} d_p(u, w) &= ((u_x - w_x)^p + (u_y - w_y)^p)^{1/p} \\ &< (u_x^p + u_y^p)^{1/p} \\ &= d_p(v, u). \end{aligned}$$

Case 2: $u_y < w_y$.
Then

$$\begin{aligned} d_p(u, w) &= ((u_x - w_x)^p + (w_y - u_y)^p)^{1/p} \\ &\le ((u_x - w_x + w_y - u_y)^p)^{1/p} \quad (*) \\ &= u_x - u_y - (w_x - w_y) \\ &< u_x - u_y \le u_x \le (u_x^p + u_y^p)^{1/p} \\ &= d_p(v, u). \end{aligned}$$

4

The inequality (*) follows from the fact that

$$(a+b)^p \geq a^p + b^p \text{ for any real numbers } p \geq 1,\ a \geq 0, \text{ and } b \geq 0. \qquad \square$$

3.2. The Total Number of all Octant Neighbours

A closest point of v in some region $R_i(v) \in \rho_p^2(v)$ is said to be an *octant neighbour* of v. We will let $\mathrm{ONG}_i(V)$ denote the graph which connects each point $v \in V$ to its octant neighbours in the octant $R_i(v)$. Further, we define the union of the above graphs as the *octant neighbour graph* of V, denoted $\mathrm{ONG}(V)$, i.e.

$$\mathrm{ONG}(V) = \bigcup_{i=1}^{8} \mathrm{ONG}_i(V).$$

In the L_∞ (and L_1) metric the RNG for a set of n points can have as many as $\Omega(n^2)$ edges (see Fig. 2). This, together with Theorem 1, means that for any narrow partition of the plane the total number of region neighbours can be $\Omega(n^2)$. This holds true especially for ONG. In $\Re_p^2$, $1 < p < \infty$, the situation is, however, different. Now the number of edges is linear for any point set, as shown in the next theorem. For the proof of the theorem we need the following lemma.

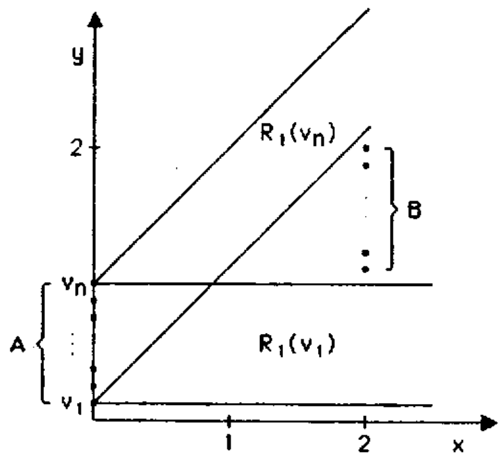

Fig. 2. A worst case set of $2n$ points in $\Re_\infty^2$, $A = \{(0, i/n) \mid i = 1, 2, \ldots, n\}$ and $B = \{(2, 1 + i/n) \mid i = 1, 2, \ldots, n\}$. Each point of B is a relative neighbour, as well as an octant neighbour in $R_1(v)$, of any point v of A

Lemma 1: *Let V be an arbitrary set of points in the plane. The graph $ONG_i(V)$, that connects every point $v \in V$ to its octant neighbours in the octant $R_i(v)$, is planar for any L_p metric, $1 < p < \infty$.*

Proof: Because of the symmetry it is sufficient to give the result for $\mathrm{ONG}_1(V)$. Actually we shall prove that $\mathrm{ONG}_1(V)$ is a planar graph even if each point is connected to its octant neighbours by a straight-line edge.

Let v be an arbitrary point in the plane, and let $q_N = q_1, q_2, \ldots, q_k = q_S$ $(1 \le k < n)$ be the octant neighbours of v in $R_1(v)$. Further, assume that the octant neighbours are ordered from "north" to "south", that is, $q_{1x} < q_{2x} < \ldots < q_{kx}$ and $q_{1y} > q_{2y} > \ldots > q_{ky}$ (see Fig. 3). Observe that in the L_p metric, $1 < p < \infty$, two or more distinct points can be equidistant from a point only if their x, as well as their y coordinates are different. Clearly, the (straight-line) edges (v, q_i) do not intersect. To conclude the proof we show that the ONG_1-edges emanating from the two points v and u do not intersect.

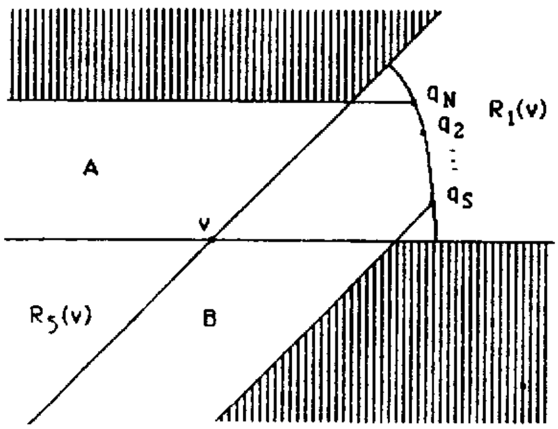

Fig. 3. Point v and its octant neighbours in $R_1(v)$

For convenience, let us assume that v is the origin. Now five different cases arise.

Case 1: $u \in R_1(v)$.

Because the points q_i, $i = 1, \ldots, k$, are the octant neighbours of v, the open circle $C_p(v, d_p(v, q_i))$, with centre v and radius $d_p(v, q_i)$, does not contain any point of V inside the region $R_1(v)$. On the other hand, if $u \in R_1(v) \setminus C_p(v, d_p(v, q_i))$ the edges of $ONG_1(V)$ adjacent to u cannot intersect the edges emanating from v, because $R_1(u) \cap C_p(v, d_p(v, q_i)) = \emptyset$. Only in the degenerate case, when u is one of the points $\{q_S, \ldots, q_N\}$, does an intersection at the end point u occur.

Case 2: u belongs to the shaded area of Fig. 3, that is,

$$u \in \{w \in \Re_p^2 \mid w_y > q_{Ny}\} \setminus R_1(v) \text{ or } u \in \{w \in \Re_p^2 \mid w_y - q_{Sy} \le w_x - q_{Sy}\} \setminus R_1(v).$$

Then, trivially, no intersections are possible between the octant neighbour edges because the edges emanating from u are totally inside the narrow region $R_1(u)$.

Case 3: $u \in R_5(v)$.

Now for any point $q \in R_1(v)$

$$\begin{aligned} d_p(u, q) &= (|u_x - q_x|^p + |u_y - q_y|^p)^{1/p} \\ &> (|u_x|^p + |u_y|^p)^{1/p} \\ &= d_p(u, v), \end{aligned}$$

because $u_x < 0$, $u_y \le 0$, $q_x \ge 0$, and $q_y > 0$. Hence, no point $q \in R_1(v)$ can be an octant neighbour of u in $R_1(u)$. This means that the ONG_1-edges emanating from u do not intersect the corresponding edges of v (except perhaps at the end point v).

Case 4: $u \in A = \{w \in \mathfrak{R}_p^2 \mid 0 < w_y \le q_{Ny}, w_x \le w_y\}$.

a) If a point $q \in \{w \in \mathfrak{R}_p^2 \mid w_x \le q_{Nx}\} \cap R_1(u)$ is an octant neighbour of u it is easily seen that the edge (u, q) cannot intersect the edges emanating from v.

b) On the other hand, a point $q \in \{w \in \mathfrak{R}_p^2 \mid w_x > q_{Nx}, w_y \ge q_{Ny}\}$ cannot be an octant neighbour of u in $R_1(u)$ because

$$\begin{aligned} d_p(u,q) &= ((q_x - u_x)^p + (q_y - u_y)^p)^{1/p} \\ &> ((q_{Nx} - u_x)^p + (q_{Ny} - u_y)^p)^{1/p} \\ &= d_p(u, q_N). \end{aligned}$$

c) Neither can a point $q \in \{w \in \mathfrak{R}_p^2 \mid w_x > q_{Nx}, w_y < q_{Ny}\}$ be an octant neighbour of u in $R_1(u)$. This becomes clear, for instance, as follows. Let r be the intersection between the edge (u, q) and the boundary of the open circle $C_p(v, d_p(v, q_N))$. This intersection must exist because the area $C_p(v, d_p(v, q_N)) \cap R_1(v)$ does not contain any point of V. Now, $d_p(u,q) \ge d_p(u,r)$. Because we always have $q_N \in R_1(u)$, it is sufficient to show that q_N is the closest point to u among the points on the boundary of the open circle, i.e.

$$d_p(u, q_N) < d_p(u,r) \text{ for all } r \in \mathfrak{R}_p^2 : r_x^p + r_y^p = q_{Nx}^p + q_{Ny}^p,\ 0 < u_x \le u_y \le r_y < q_{Ny} < q_{Nx}.$$

To prove this let us consider the p-th power of the distance $d_p(u,r)$:

$$S(r_x) = d_p(u,r)^p = (r_x - u_x)^p + (r_y - u_y)^p, \text{ where } r_y = (q_{Nx}^p + q_{Ny}^p - r_x^p)^{1/p}.$$

Now

$$\begin{aligned} \frac{dS}{dr_x} &= p(r_x - u_x)^{p-1} + p(r_y - u_y)^{p-1} \cdot 1/p \cdot (q_{1x}^p + q_{1y}^p - r_x^p)^{1/p-1} \cdot (-p\, r_x^{p-1}) \\ &= p\,[(r_x - u_x)^{p-1} - (r_y - u_y)^{p-1} \cdot (r_x/r_y)^{p-1}]. \end{aligned}$$

Since $(r_x - u_x)/(r_y - u_y) > r_x/r_y$ on that part of the boundary of the open circle in question, we have $dS/dr_x > 0$, and thus $d_p(u,r) > d_p(u, q_N)$.

Case 5: $u \in B = \{w \in \mathfrak{R}_p^2 \mid w_y < 0, w_x > w_y, w_y - q_{Sy} > w_x - q_{Sx}\}$.

a) If a point $q \in \{w \in \mathfrak{R}_p^2 \mid w_y \le q_{Sy}\} \cap R_1(u)$ is an octant neighbour of u, it is evident that the edge (u, q) cannot intersect any of the ONG_1-edges emanating from v.

b) A point $q \in \{w \in \mathfrak{R}_p^2 \mid w_x > q_{Sx}, w_y \ge q_{Sy}\} \cap R_1(u)$ cannot be an octant neighbour of u because

$$\begin{aligned} d_p(u,q) &= ((q_x - u_x)^p + (q_y - u_y)^p)^{1/p} \\ &> ((q_{Sx} - u_x)^p + (q_{Sy} - u_y)^p)^{1/p} \\ &= d_p(u, q_S). \end{aligned}$$

c) Suppose that $q \in \{w \in \mathfrak{R}_p^2 \mid w_x > q_{Sx}, w_y < q_{Sy}\} \cap R_1(u)$. As in Case 4 c, it is enough to consider the distances $d_p(u,r)$ for the points r on the boundary of the open circle $C_p(v, d_p(v, q_S))$ in the region $R_1(u)$, i.e.

$$d_p(u, q_S) < d_p(u, r) \text{ for all } r \in \Re_p^2 : r_x^p + r_y^p = q_{Sx}^p + q_{Sy}^p,\ u_y < 0,\ u_x > u_y,\ r_x > r_y > q_{Sy}.$$

Denote $S(r_x) = d_p(u, r)^p$, then $dS/dr_x < 0$ in the area concerned. Hence we have $d_p(u, q_S) < d_p(u, r)$. □

The above lemma has two direct consequences.

Theorem 3: *The number of edges in the octant neighbours graph $ONG(V)$ is $\theta(n)$ for a set V of n points in $\Re_1^2$, $1 < p < \infty$.*

Proof: By Euler's formula a planar graph $\mathrm{ONG}_i(V)$ can have at most $3n - 6$ edges ($n \geq 3$), and hence $\mathrm{ONG}(V)$ has at most $24n - 48$ edges. The $\Omega(n)$ lower bound is trivial. □

Theorem 4: *The relative neighbourhood graph $RNG(V)$ contains $\theta(n)$ edges for a set V of n points in $\Re_p^2$, $1 < p < \infty$.*

Proof: The upper bound follows directly from the Theorems 1 and 2, which showed that $\mathrm{ONG}(V) \supseteq \mathrm{RNG}(V)$. Since the relative neighbourhood graph is a supergraph of a minimum spanning tree for any distance measure (see [10, Chapter 1]), $\Omega(n)$ is a lower bound for the number of edges in $\mathrm{RNG}(V)$. □

The last theorem generalizes the results of Toussaint [20] concerning the Euclidean case. Observe, however, that Theorem 4 can be proved much more directly by showing that the relative neighbourhood graph is planar for any set of points in $\Re_p^2$, $1 < p < \infty$. This also gives a tighter upper bound for the number of edges in the RNG.

4. Algorithms Based on the Region Approach

Here we give two RNG-algorithms based on the partition studied in Section 3. First, a simple method is introduced that runs in time $\theta(n^2)$ for any L_p metric, $1 < p < \infty$. The method is then refined for the L_1 and L_∞ metrics. If m denotes the number of edges in the output, the refined algorithm has a running time of $\theta(n \log n + m)$. The time bound is asymptotically optimal.

4.1. *An Algorithm in the L_p Metric, $1 < p < \infty$*

Consider the partition $\rho_p^2 = \{R_1, \ldots, R_8\}$ of the plane, and the octant neighbour graph induced by the partition. By Theorem 1 the relative neighbours of a point v are also octant neighbours of v in an octant $R_i(v)$. This suggests that to compute the relative neighbourhood graph for a point set V, the following straightforward method can be used: for each point v of V, search for the octant neighbours of v, and then check whether the corresponding lunes include any point of V. The search for the octant neighbours of v can be made in $\theta(n)$ time, by determining for each point u of $V \setminus \{v\}$ the octant in which u is located with regard to v, and then by updating when necessary the tentative closest point(s) of the octant. Each of the lune tests demands

$\theta(n)$ time if done with brute force. Hence, the running time of this simple algorithm is $\theta(n^2+Xn)$, where X denotes the total number of edges in the octant neighbour graph ONG(V). Combining this with Theorem 3 we have

Theorem 5: *The running time of the simple RNG-algorithm based on the region approach is* $\theta(n^2)$ *in* $\Re_p^2$, $1<p<\infty$.

4.2. An Algorithm in the L_1 and L_∞ Metrics

For the L_∞ (and L_1) metric, the number of edges in ONG can be quadratic on the number of points. Hence, the worst-case running time of the straightforward RNG-algorithm is $\theta(n^3)$. However, on the assumption that for every point its octant neighbour is unique in each octant, the RNG can be computed in $\theta(n\log n)$ time (compare [15]). This is because first the unique octant neighbours can be found in $\theta(n\log n)$ time (see [7] or [8]) and then the "emptiness" of the $\theta(n)$ lunes, which are now rectangular boxes, can each be tested in $\theta(\log n)$ time, by using a (static) range searching data structure demanding only $\theta(n\log n)$ preprosessing (see [7] or [29]). Even when the uniqueness of the octant neighbours is not assumed these ideas can be utilized, as will be shown below. (Because of the isometry between the spaces $\Re_1^2$ and $\Re_\infty^2$, hereafter only the L_∞ metric is considered.)

In the L_∞ metric the most significant property is that all octant neighbours of v in $R_i(v)$ lie on the same line-segment. For the sake of simplicity, let us consider only the octant neighbours of v in the first octant $R_1(v)$. We assume that the octant neighbours $q_N=q_1, q_2, \ldots, q_k=q_S$ are ordered according to their y-coordinate values from "north" to "south". Then lune $(v, q_{i+1}) \supset$ lune (v, q_i), $i=1,2,\ldots,k-1$, i.e. lune(v, q_{i+1}) contains all the previous lunes such that the difference between two lunes is always a rectangle (see Fig. 4) This observation gives us directly a method for

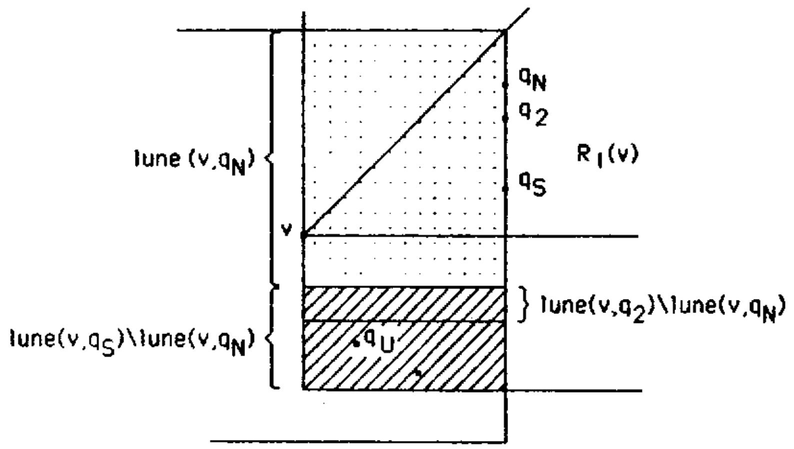

Fig. 4. The octant neighbours of v and the corresponding lunes in $\Re_\infty^2$

checking which of the octant neighbours are also relative neighbours. First, perform a range query to check whether lune(v, q_N) contains any point of V. If it does not, none of the octant neighbours is a relative neighbour; if it does, find the northernmost point inside the rectangle lune$(v, q_S)\setminus$lune(v, q_N). This point (if any) determines directly which of the region neighbours are also relative neighbours.

We shall now present an efficient algorithm to implement the above observation. First we show how to compute the relative neighbours of any point v in the region $R_1(v)$. The other seven regions are handled in the same way. In fact, by using appropriate linear transformations of the plane the subroutines for the region R_1 can be applied as such to find the relative neighbours in all of the eight regions (see, e.g. [8]).

The algorithm begins by sorting all points in x as a primary-key, using y as a secondary-key. The sorted double-linked list of points is needed for outputting. For each point we maintain a pointer to its position in the list. All this can be done in $\theta(n \log n)$ time, where n is the number of points. When the sorted list is available there is no need to find all octant neighbours of the point v; instead, we find only its northernmost and southernmost octant neighbours in $R_1(v)$. The search for the two neighbours of each of the n points can be accomplished in $\theta(n \log n)$ time. For this purpose we can use, for example, the divide-and-conquer algorithm proposed by Guibas and Stolfi [8]. Basically their algorithm computes all northernmost octant neighbours, but the algorithm can easily be adapted so that with it both the northernmost and the southernmost octant neighbours can be found simultaneously. This adaptation is left to the reader. (Actually, the main subroutine of the algorithm of Guibas and Stolfi computes the nearest L_1 neighbour in the first quadrant for each point, but also here linear transformations of the plane can be used to map the nearest octant L_∞ neighbours to nearest first quadrant L_1 neighbours.)

After finding the northernmost octant neighbour q_N and the southernmost octant neighbour q_S of a point v, we have to perform two kinds of range queries: first one to check whether lune (v, q_N) contains any point of V and second one to search for the nearest point of the upper boundary of the rectangle lune$(v, q_S)\setminus$lune(v, q_N). Of course, a data structure that solves the second type of query efficiently can also be used to check the emptiness of a box. The range searching problem has been widely studied (see, e.g. [4, 7, 29]). The basic problem with efficient data structures for range searching is that their space requirements are non-linear. However, in our case the order of the queries is totally unimportant, which means that the queries can be answered off-line. Hence, we can apply the finding of Edelsbrunner and Overmars [6] (or [17, Chapter 8]) that in a batched environment the range searching problem can be solved also space-efficiently.

Theorem 6 [6]: *Let n denote the number of points in the set, n_q the number of range queries, and F the total number of points found in response to the queries. Then the batched static version of the two-dimensional range searching problem can be solved in time $O((n+n_q)\log n + n_q \log n_q + F)$ and the amount of storage required is $O(n+n_q)$.*

In our case the total number of range queries is $2n$. Further,, the total number of answers is $O(n)$ because we are returning only one point, the northernmost point, per query. Hence, the range queries in our RNG-algorithm can be performed in $O(n \log n)$ time.

Now if lune(v, q_N) contains a point of the set V, none of the octant neighbours of v is a relative neighbour of v. Otherwise, if lune(v, q_N) is empty, the point q_U (if any) determines the relative neighbours of v. That is to say, all those octant neighbours q_i for which $d_\infty(v, q_i) \leq d_\infty(q_i, q_U)$ are also relative neighbours of v. It will easily be seen that the outputting of all relative neighbours can be done in time proportional to the number of edges in the RNG. This is achieved by traversing the presorted list of points, starting from q_N, until the first point with a different x-coordinate is met, or until a point q_i is met for which $d_\infty(q_i, q_U) < d_\infty(v, q_N)$.

Thus, we have proved

Theorem 7: *The relative neighbourhood graph for a set of n points in* $\Re^2_\infty$ *can be constructed in* $\theta(n \log n + m)$ *time, where m is the size of the output, i.e. the number of edges in the relative neighbourhood graph.*

As shown in Fig. 2 the number of RNG-edges can be $\Omega(n^2)$ in the worst case. However, for a point set possessing only a few points with the same x-coordinate values, as well as the same y-coordinate values, the size of the output is almost linear and then the time needed for outputting is covered by the other computation.

The algorithm is also optimal within a multiplicative constant. Trivially, any RNG-algorithm has to perform the outputting, but in addition to this, an RNG-algorithm can be used to sort a one-dimensional point set [19] from which the lower bound follows. The optimality is valid in all those models of computation where sorting demands $\Omega(n \log n)$ time.

5. Conclusions

In this paper it has been shown that the region approach can be used successfully to compute relative neighbourhood graphs in L_p metrics. In the two-dimensional space with the L_p metric, $1 < p < \infty$, we described a very simple algorithm running in $\theta(n^2)$ time for any set of n points. In the L_∞ and L_1 metrics, the method could be refined to a $\theta(n \log n + m)$ algorithm, where m is the size of the output. The latter algorithm is also asymptotically optimal.

Some interesting questions still remain unanswered.

1. Can one prove tighter bounds for the number of edges in RNG or ONG? Urquhart [27] has considered this problem for RNG in the Euclidean plane; for ONG the work is yet to be done.
2. Would it be possible to find all octant neighbours faster than in $\theta(n^2)$ time? Further, could one reduce ONG to RNG more rapidly than in time $\theta(n^2)$ obtained by the brute force reduction? If the answer to both questions is positive, this will immediately make it possible to improve also the RNG-algorithm based on the region approach in the L_p metric, $1 < p < \infty$.

3. How are the given algorithms to be extended to higher dimensions? Supowit [19] proposed one solution for Euclidean case, but would it be possible to devise computationally a more efficient partition of the space into narrow regions? In practice, the modification of Urquhart's RNG-algorithm, introduced in [12], compares favourably with the region approach because of its simplicity.

Acknowledgements

I gratefully acknowledge the help and support of Olli Nevalainen. This research was sponsored by the Ministry of Education and the Academy of Finland.

References

[1] Bentley, J. L., Friedman, J. H.: Data structures for range searching. Comput. Surveys *11*, 397–405 (1979).

[2] Bentley, J. L., Maurer, H. A.: Efficient worst-case data structures for range searching. Acta Inform. *13*, 155–168 (1980).

[3] Bentley, J. L., Weide, B. W., Yao, A. C.: Optimal expected-time algorithms for closest point problems. ACM Trans. Math. Software *6*, 563–580 (1980).

[4] Chazelle, B.: Filtering search: a new approach to query-answering. SIAM J. Comput. *15*, 703–724 (1986).

[5] Coppersmith, D., Lee, D. T., Wong, C. K.: An elementary proof of nonexistence of isometries between l_p^k and l_q^k. IBM J. Res. Develop. *23*, 696–699 (1979).

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

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

[8] Guibas, L. J., Stolfi, J.: On computing all north-east nearest neighbors in the L_1 metric. Inform. Process. Lett. *17*, 219–223 (1983).

[9] Jaromczyk, J., Kowaluk, M.: A note on relative neighbourhood graphs. Proc. 3rd Annual Symposium on Computational Geometry, 1987, pp. 233–241.

[10] Katajainen, J.: Bucketing and Filtering in Computational Geometry. Report A46, Department of Computer Science, University of Turku, Turku, Finland, 1987.

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

[12] Katajainen, J., Nevalainen, O.: An almost naive algorithm for finding relative neighbourhood graphs in L_p metrics. Informatique Théorique et Applications/Theoretical Informatics and Applications *21*, 199–215 (1987).

[13] Katajainen, J., Nevalainen, O., Teuhola, J.: A linear expected-time algorithm for computing planar relative neighbourhood graphs. Inform. Process. Lett. *25*, 77–86 (1987).

[14] Lee, D. T.: Two-dimensional Voronoi diagrams in the L_p metric. J. Assoc. Comput. Mach. *27*, 604–618 (1980).

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

[16] O'Rourke, J.: Computing the relative neighborhood graph in the L_1 and L_∞ metrics. Pattern Recognition *15*, 189–192 (1982).

[17] Overmars, M. H.: The Design of Dynamic Data Structures. Lecture Notes in Computer Science 156. Heidelberg: Springer-Verlag 1983.

[18] Preparata, F. P., Shamos, M. I.: Computational Geometry: An Introduction. Heidelberg: Springer-Verlag 1985.

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

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

[21] Toussaint, G. T.: Pattern recognition and geometrical complexity. Proc. 5th International Conference on Pattern Recognition, 1980, pp. 1324–1347.

[22] Toussaint, G. T.: Decomposing a simple polygon with the relative neighborhood graph. Proc. 18th Annual Allerton Conference on Communication, Control, and Computing, 1980, pp. 20–28.

[23] Toussaint, G. T.: Comment on "Algorithms for computing relative neighbourhood graph". Electron Lett. *16*, 860–861 (1980).

[24] Toussaint, G. T., Bhattacharya, B. K., Poulsen, R. S.: The application of Voronoi diagrams to nonparametric decision rules. Proc. Computer Science and Statistics: 16th Symposium on the Interface, 1984.

[25] Toussaint, G. T., Menard, R.: Fast algorithms for computing the planar relative neighborhood graph. Proc. 5th Symposium on Operations Research, 1980, pp. 425–428.

[26] Urquhart, R. B.: Algorithms for computation of relative neighbourhood graph. Electron Lett. *16*, 556–557.

[27] Urquhart, R. B.: Some properties of the planar Euclidean relative neighbourhood graph. Pattern Recognition Lett. *1*, 317–322 (1983).

[28] Urquhart, R. B.: Some New Techniques for Pattern Recognition Research and Lung Sound Signal Analysis. Ph. D. Thesis, Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow, Scotland, 1983.

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

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

Jyrki Katajainen
Department of Computer Science
University of Turku
SF-20500 Turku
Finnland

Printed in Austria

Druck: Paul Gerin, A-1021 Wien