Efficient Algorithms for Computing the β-Spectrum

S. V. Rao
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati, India
E-mail: svrao@iitg.ernet.in

Asish Mukhopadhyay
School of Computer Science
University of Windsor, Canada
E-mail: asishm@cs.uwindsor.ca

Abstract

Kirkpatrick and Radke [KR85] proposed the notion of a β-skeleton as a measure of the "internal shape" of a planar set of points. We get an entire spectrum of shapes by varying the parameter β. For a fixed value of β, the "internal shape" is a geometric graph obtained by joining each pair of points whose β-neighborhood is empty.

For $\beta \geq 1$, this neighborhood of a pair of points p_i, p_j is the **interior** of the intersection of two circles of radius $\beta|\overline{p_i p_j}|/2$, centered at the points $(1-\beta/2)p_i + (\beta/2)p_j$ and $(\beta/2)p_i + (1-\beta/2)p_j$, respectively. For $\beta \in [0,1]$, it is the **interior** of the intersection of two circles of radius $|\overline{p_i p_j}|/(2\beta)$, passing through p_i and p_j.

In this paper we propose efficient algorithms for computing the entire spectrum of β-shapes of a finite planar point set in the L_2 metric.

We also present algorithms for computing the spectra of of $k\beta$-shapes and additively-weighted β-shapes.

1 Introduction

The concept of a β-skeleton was suggested as a way of capturing the "internal shape" of a planar set of points, an issue that arises in areas such as **Computational Morphology** [Tou88] and **Pattern Recognition** [DH73]. Other notable shape descriptors are α-shape [EM94] and γ-skeleton [Vel92].

Indeed, we get an entire spectrum of shapes by varying the parameter β. For a fixed value of β, the "internal shape" is a geometric graph, obtained by joining each pair of points whose β-neighborhood is empty.

Thouugh it is debatable in what sense a geometric graph defined on a point set captures its "shape", these graphs have proved very useful in the study of various geometric optimization problems [CTL92a], [CTL92b], [Mir93]. This provides another motive, and the one that we shall emphasize in this paper, for finding efficient algorithms for computing these "shape" graphs.

For $\beta \geq 1$, the β-neighborhood of a pair of points p_i, p_j is the **interior** of the intersection of two circles of radius $\beta|\overline{p_i p_j}|/2$, centered at the points $(1-\beta/2)p_i + (\beta/2)p_j$ and $(\beta/2)p_i + (1-\beta/2)p_j$, respectively. For $\beta \in (0,1]$, it is the **interior** of the intersection of two circles of radius $|\overline{p_i p_j}|/(2\beta)$, passing through p_i and p_j. We denote this neighborhood by $lune_\beta(p_i,p_j)$. A point that lies in this neighbourhood is a *witness* to this edge. The parameter β characterizes the neighborliness of the pair of points.

The algorithmic problem of computing a β-skeleton has been addressed, among others, by Lingas [Lin94], Jaromczyk *et al* [JKY]. They have proposed a linear-time algorithm for constructing a β-skeleton in the range $1 \leq \beta \leq 2$ under the distance-metric L_p $(1 < p < \infty)$, provided $\beta = 2$ or $p = 2$. Recently, Rao and Mukhopadhyay [RM97] have proposed efficient algorithms for computing a β-skeleton, without any restriction on p or β.

In this paper we propose efficient algorithms for computing the entire spectrum of β-shapes of a planar set of points P in the L_2 metric. This means computing for each pair of points the largest β value, β_{max}, for which the β-neighborhood of the pair is empty. The motives are algorithmic: first, this is an interesting problem for which no efficient solution was known before; second, this gives a compact representation of all the β-shapes. Our algorithms are in $O(n^{3/2}\log n + K)$ time for $\beta_{max} \geq 1$ and in $O(n^2 \log n + K)$ time for $\beta_{max} \geq 0$. A point that lies in the β-neighborhood of an edge is a *witness* to this edge. The quantity K is a count of the number of times that the edges of the initial (Delaunay or complete) graph on P are witnessed for $\beta = \infty$.

The rest of the paper is organized as follows. In the next section we present algorithms based on the techniques of range searching and sweepline. In the next two sections we discuss respectively how to compute efficiently the spectra of $k\beta$-shapes and additively weighted β-shapes. We conclude in the fifth and final section.

2 Computing β-spectrum

The main idea is to compute the witness set, W_e, for each edge e of a super-graph of all the β-skeletons. For

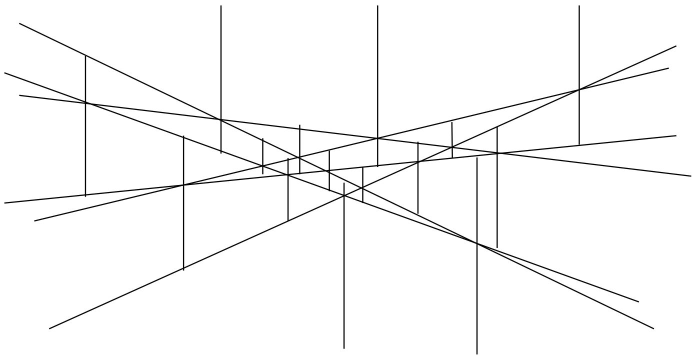

Figure 1: New planar subdivision $\mathcal{A}'$

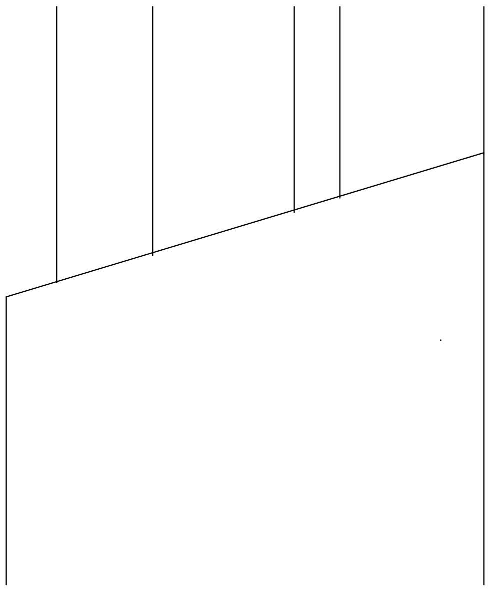

Figure 2: Complex trapezium.

$\beta \geq 1$ we start with Delaunay triangulation on P since it is a super-graph of the 1-skeleton, while for $\beta \geq 0$, we start with the complete graph on P. We discuss only the first case as the second case can be handled similarly.

A key observation is that all possible witnesses of an edge are contained in the lune that corresponds to $\beta = \infty$. This is an open infinite strip bounded by the lines normal to the edge at its end-points. We assume without loss of generality that no edge is horizontal, since all such edges can be handled separately in a preprocessing step in $O(n \log n)$ time.

Under the point-line duality transformation T, an infinite strip in the primal plane maps to a vertical line segment in the dual plane. So a point $p \in W_e$ if and only if T_p intersects T_e. The problem of computing W_e for an edge e, therefore, reduces to computing all the lines that intersect a vertical line segment corresponding to e in the dual plane.

We consider two different approaches to the problem.

2.1 The Range-Searching Approach

In this approach we preprocess the arrangement, $\mathcal{A}$, in two steps so that, given a vertical query segment q, we can report the lines of L that intersect q in $O(\log n + w_e)$ time, where w_e is the number of intersecting lines.

We first trapezoidize the the arrangement, $\mathcal{A}$. The resulting planar subdivision, $\mathcal{A}'$, is shown in Fig. 1. This is done in $O(n^2 \log n)$ time by a sweep.

A trapezium in $\mathcal{A}'$ can have many vertices on its non-parallel sides (see Fig. 2). It has an upper (resp. lower) anomaly if its upper (resp. lower) side has more than three vertices. In the second step, we remove these anamolies.

For an edge $e \in \mathcal{A}$, let $A = \{a_1, a_2, \ldots, a_u\}$ be the set all vertical edges in $\mathcal{A}'$ that are incident on e from above and $B = \{b_1, b_2, \ldots, b_v\}$ be the set of those edges that are incident from below.

For j in $[1 \ldots v-1]$ let $\{a_i, a_{i+1}, \ldots, a_j\}$ be the subset

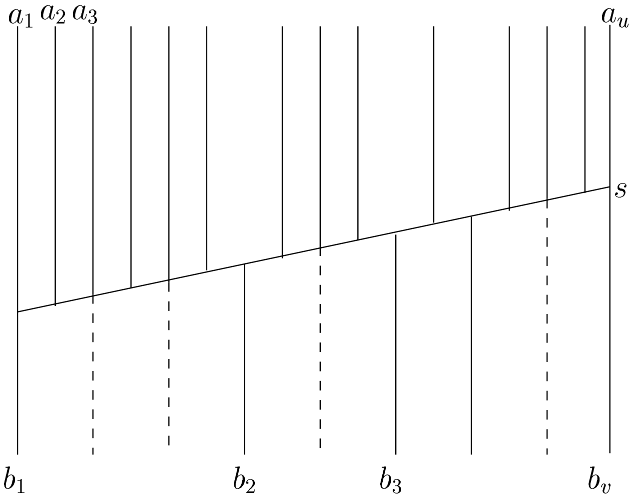

Figure 3: Removing upper anomaly

of A whose end-points on e lie strictly between those of b_j and b_{j+1}. We extend the edges $a_{i+1}, a_{i+3}, \ldots, a_{i+l}$, where $l \leq (j - i)$ (see Fig. 3) up to an edge of $\mathcal{A}$ lying immediately below e. While this removes the upper anomalies of all the trapezium that hang on e, it may create upper anomalies for the ones that lie immediately below.

To handle all the regions uniformly, we process $\mathcal{A}'$ by levels of the original arrangement $\mathcal{A}$ from the topmost level to the bottommost one, removing all upper anomalies. And again from the bottom to the top, removing all lower anomalies. The resulting planar subdivision $\mathcal{A}''$ is called a hive-graph H [Cha86].

The time and space complexities of this preprocessing step are in $O(n^2 \log n)$ and $O(n^2)$ respectively.

We preprocess $\mathcal{A}''$ for point location in $O(n^2 \log n)$ time [EGS86]. We index the regions of $\mathcal{A}''$, and store these regions in an array R such that the array index of a region is identical to its region index. Along with a region we also store the index of the region just above it. Given a query segment q, we first locate its bottom end-point in $\mathcal{A}''$ in $O(\log n)$ time; we then use its region index to locate it in the array R and determine the index of the region just above. We repeat this till we hit the top end-point of q. We claim the following result.

Lemma 2.1 *A set of n lines can be preprocessed in $O(n^2 \log n)$ time using $O(n^2)$ space such that given a vertical query segment we can report all lines that intersect it in $O(\log n + w_e)$ query time, where w_e is the size of the output.*

It follows from the above lemma that if we are given a set of n lines L and a set of $O(n)$ vertical line segments V, the intersections of the lines in L with the line segments in V can be computed in $O(n^2 \log n + K)$ amortized time, where K is a count of all such intersections.

We can whittle away a $\sqrt{n}$ factor from the above complexity by preprocessing groups of lines of size $\sqrt{n}$. The details are in the lemma below.

Lemma 2.2 *Given a set L of n lines and a set V of $O(n)$ vertical segments, computing the intersections of the vertical segments and the lines in L can be done in $O(n^{3/2} \log n + K)$ time using $O(n)$ space, where K is a count of all such intersections.*

Proof: We divide the problem into m subproblems, each consisting of n/m lines and all the vertical segments in V. Solve each subproblem by the algorithm above. Summing up the time complexities of the individual subproblems, we find that the optimal time complexity of $O(n^{3/2} \log n + K)$ is achieved for $m = \sqrt{n}$. The space required is $O(n)$ since we are considering $\sqrt{n}$ lines at a time. ∎

Knowing the potential witness set of each edge in DT, we can compute the β_{max} values for each edge e in time proportional to the size of W_e. Therefore β_{max} value for all the edges of DT can be computed in time proportional to K. Thus we have the following result.

Theorem 2.3 *The β-spectrum of a set of points P for $\beta \geq 1$ can be computed in $O(n^{3/2} \log n + K)$ time using $O(n)$ space, where K is a count of the total number of times edges in the DT are witnessed for $\beta = \infty$.*

Similarly, we can obtain the following result.

Theorem 2.4 *The β-spectrum of a set of n points P for $\beta \geq 0$ can be computed in $O(n^2 \log n + K)$ time and $O(n^2)$ space, where K is a count of the total number of times the edges of the complete graph are witnessed for $\beta = \infty$.*

We note that in this case we have to query $O(n^2)$ vertical line segments and hence dividing the lines into groups of size $\sqrt{n}$ does not help.

2.2 Sweepline Approach

In this alternate approach, we take advantage of the fact that we know all the query segments beforehand.

In the **sweepline status** $\mathcal{L}$ we store the lines in L according to a total ordering of their intersections. The **event queue** Q is an union of the set of vertical segments V and the intersections of the lines in L, sorted by their x-values.

The algorithm works as follows. As we sweep the structure, if the next event corresponds to an intersection of two lines we update $\mathcal{L}$ by reordering their positions in the intersection-order with the sweepline. If the event is a vertical line segment, we report all the lines in $\mathcal{L}$ that intersect it. This can be done in $O(\log n + w_e)$ time, where w_e is the number of lines that intersect the vertical line segment.

The same two theorems of the previous section also hold.

3 $k\beta$-Spectrum

A natural generalization of the notion of a β-skeleton to that of a $k\beta$-skeleton is obtained by allowing a β-lune to contain up to $k-1$ points. In particular, for $\beta = 1$ we get the k-Gabriel graph which is a subgraph of the k-Delaunay graph, and can be computed in $O(k^2 n \log n)$ time [SC90].

For for $\beta = 2$, we get the k-Relative neighborhood graph (k-RNG). For fixed k, Chang *et al.* [CTL92b] presented an $O(n^2)$ algorithm to compute this graph. Later, Su and Chang improved this to $O(n^{5/3} \log n)$ [SC91].

Chang *et al* have found a surprising connection between k-relative neighbourhood graphs and some problems in graph algorithms [CTL92b], [CTL92a], [CTL91].

Similarly, a **$k\beta$-spectrum** generalizes the notion of a β-spectrum. It is a weighted geometric graph G. The weight of an edge is the largest β value, β_{max}, for which the β-neighborhood of the edge contain at most $k-1$ points.

To compute the spectrum for $\beta \geq 1$, we start with the k-Delaunay graph (k-DG) [SC90], known to be a super graph of a $k\beta$-skeleton. To compute the spectrum for β values greater than or equal to 0, we start with the complete graph.

Similar to previous methods we compute the set of witnesses W_e for each edge $e \in k$-DG. For each witness $w \in W_e$ compute the β value at which w becomes witness of the edge e. The β_{max} of the edge e is the kth smallest of these β values, if $|W_e| \geq k$ otherwise $\beta_{max} = \infty$. Therefore, we claim the following result.

Theorem 3.1 *The $k\beta$-spectrum of a set P of n points in the plane for $\beta \geq 1$, can be constructed in $O(kn^{3/2} \log n + K)$ time using $O(n)$ space, where K is a count of the total number of times the edges in k-DT are witnessed for $\beta = \infty$.*

Similarly one can establish the following theorem.

Theorem 3.2 *The $k\beta$-spectrum of a set P of n points in the plane for $\beta \geq 0$, can be constructed in $O(n^2 \log n + k^2 n \log n + K)$ time using $O(n^2)$ space, where K is a count of the total number of times the edges of the complete graph are witnessed for $\beta = \infty$.*

4 Weighted β-Spectrum

In this section, we generalize the notion of β-shapes to additively weighted β-shapes. We assign a non-negative weight w_i to each point $p_i \in P$. The weight w_i expresses the power of the point p_i in the sense that each weighted point p_i can be treated as a circle B_i, centered at p_i, with radius equal to the weight w_i. Thus the weighted distance $d_w(p_i, x)$, between a point $p_i \in P$ and an arbitrary point x, is $d(p_i, x) - w_i$ if x is outside B_i, and zero otherwise. The weighted distance $d_w(p_i, p_j)$, between any two points $p_i, p_j \in P$ is the distance between B_i and B_j. Let D denote the set of circles B_i.

For all $p_i, p_j \in P$ such that B_i and B_j are disjoint, the weighted neighborhood, $lune_\beta(p_i, p_j)$, is a region lying between the circles B_i and B_j and is defined as follows: for $\beta \geq 1$, it is the interior of the intersection of two circles of radii $\beta d_w(p_i, p_j)/2 + w_i(\beta - 1)$ and $\beta d_w(p_i, p_j)/2 + w_j(\beta - 1)$, whose centers are on the line joining p_i and p_j, and which touch externally the circles B_j and B_i respectively (see Fig. 4); For $\beta \in (0, 1]$, it is the interior of the intersection of two circles of radius $d_w(p_i, p_j)/2\beta$, passing through p'_i and p'_j which are a pair of closest points on the circles B_i and B_j respectively (see Fig. 4). If the circles B_i and B_j intersect, then the $lune_\beta(p_i, p_j)$ is undefined for all $\beta \geq 0$.

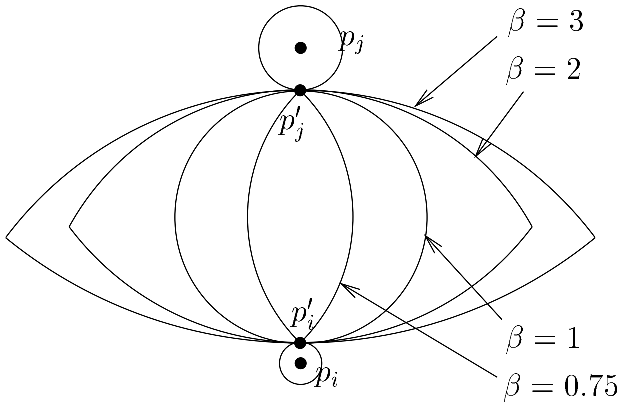

Figure 4: Neighborhood of weighted β-skeleton.

The weighted β-skeleton of P in this case is a geometric graph (P, E) such that for every $p_i, p_j \in P$, the edge $\overline{p_i p_j} \in E$ if and only if $lune_\beta(p_i, p_j)$ is not intersected by any circle $B_k \in D - \{B_i, B_j\}$.

When $\beta = 1$, the weighted β-skeleton is exactly the weighted Gabriel graph; when $\beta = 2$, we get the weighted relative neighbourhood graph [Mir93] for a pairwise disjoint set of circles D. Thus a weighted β-skeleton, for $\beta \geq 1$, is a subgraph of the WDG.

Mirzaian [Mir93] has found an interesting connection between weighted RNG and Eucidean matching.

In the rest of this section we discuss an algorithm for computing the weighted β-spectrum.

4.1 Computing weighted β-spectrum

To compute the spectrum for $\beta \geq 1$, we start with the weighted Delaunay graph (WDG) [Mir93]. For $\beta \geq 0$, we start with the complete graph. We discuss only the first case.

The previous algorithms do not generalize in a straight forward manner, because a circle in the primal plane does not have a dual that is a simple object. A simple observation, however, helps us get around this problem. The symbol $S_{\overline{p_i p_j}}$ denotes the $lune_\infty(p_i, p_j)$; we will let S_k^e denote the smallest infinite strip which encloses the weighted circle B_k and is parallel to S_e.

Observation 4.1 *A point p_k is witness to an edge e for some value of β if and only if S_k^e intersects S_e.*

Letting I_k^e and I_e denote the the duals of these strips, we have the follwing equivalent claim in the dual plane.

Observation 4.2 *A point p_k is witness to an edge e for some value of β if and only if I_k^e intersects I_e.*

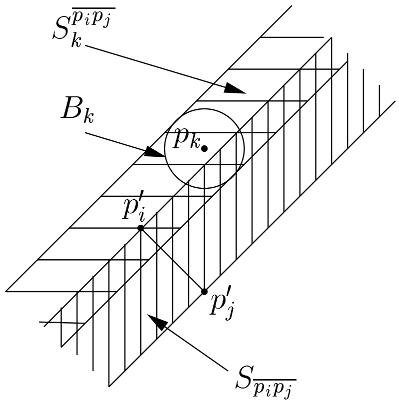

Figure 5: Illustration of Observation 1.

Therefore, to compute witness set of an edge $e \in$ WDG, we find all the vertical line segments I_k^e that intersect the vertical segment I_e in the dual plane. The following observations suggest an efficient method to do the computation.

Observation 4.3 *An x-value in the dual plane corresponds to a slope in the primal plane.*

Observation 4.4 *For a given edge $e \in$ the WDG, the vertical segments I_k^e, for $1 \le k \le n$, form overlapping intervals, lying on a line parallel to the y-axis.*

The main idea is to sweep the dual plane along the x-axis and maintain n intervals in the sweepline status that correspond to n weighted circles in the primal plane. When the sweepline $\mathcal{L}$ coincides with I_e, we report all the intervals in the sweepline status that intersect I_e. We discuss the sweep $x = 0$ to $x = \infty$. The algorithm is completed by another sweep from $x = 0$ to $x = -\infty$.

The sweepline $\mathcal{L}$ contains a relative ordering of the $2n$ end-points of n intervals by their x-values. Two adjacent end points in $\mathcal{L}$ represent an *elementary interval.* With each elementary interval EI we associate a list of intervals C_{EI}, that contain this elementary interval. We initialize $\mathcal{L}$ at $x = 0$.

As the sweep proceeds the relative ordering of these end points change and hence also the elementary intervals (see Fig. 6).

We illustrate how the sweepline status changes by way of an example: let q_1, q_2, q_3 and q_4 be four consecutive end-points in $\mathcal{L}$ (see Fig. 7). The elementary intervals EI_1, EI_2, and EI_3 respectively correspond to $[q_1, q_2], [q_2, q_3]$, and $[q_3, q_4]$.

Let the event be a change in the order of q_2 and q_3 in $\mathcal{L}$. Let q_2 be an end-point of I_i^x and q_3 of I_j^x. The new elementary intervals EI_1, EI_2, and EI_3 are respectively $[q_1, q_3], [q_3, q_2]$, and $[q_2, q_4]$. The corresponding interval lists C_{EI_1}, C_{EI_2}, and C_{EI_3} are updated as follows: if q_2

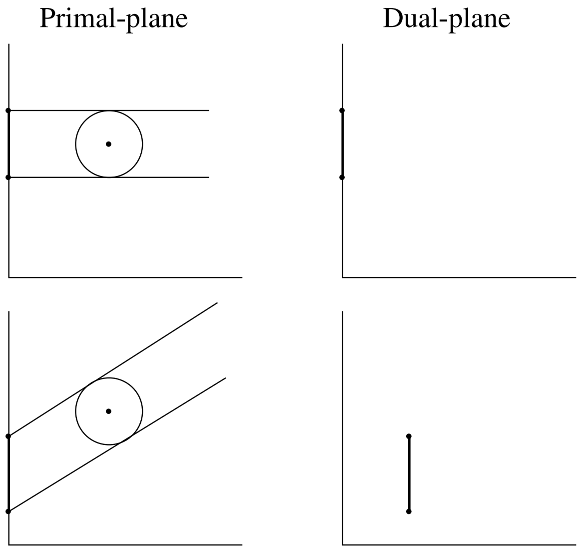

Figure 6: Rotation of a strip I_i and its effect in dual plane.

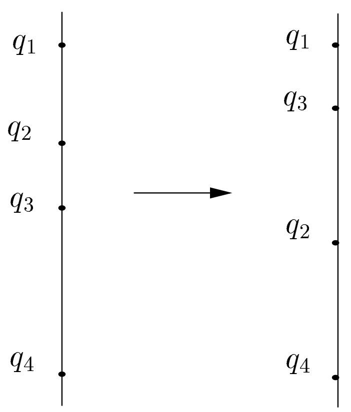

Figure 7: Event updation.

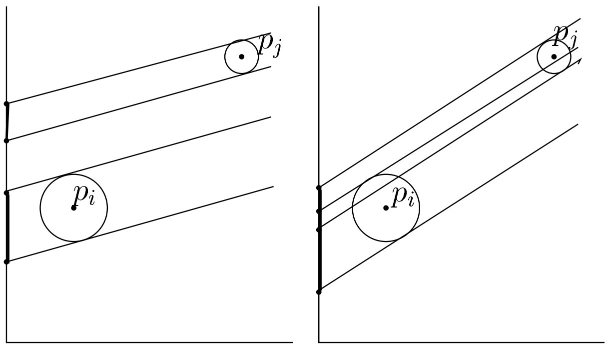

Figure 8: Finding an event.

is the top end point of I_i^x, delete I_i^x from C_{EI_2}. If it is the bottom end point of I_i^x insert I_i^x in C_{EI_2}. Similarly, update C_{EI_2} with respect to the point q_3. The interval lists C_{EI_1} and C_{EI_3} are unchanged.

When the sweepline $\mathcal{L}$ coincides with I_e, we report all the intersecting intervals as follows. We locate the top end point of I_e in an elementary interval $EI \in \mathcal{L}$. The segment I_e intersects all the intervals in C_{EI}. This acciunt for all intervals whose top end point is above EI. But the top end-points of some intersecting intervals may start below that of EI. These can be reported by going down from the top end-point to the bottom end-point of I_e in the sweepline status $\mathcal{L}$. During this, if the top end-point of any interval is encountered we report the interval corresponding to this top end-point.

It is easy to determine the events arising out of the adjacency of a pair of points on the sweepline status $\mathcal{L}$. This is the x-value corresponding to the slope of a common tangent to the weighted circles (in the primal plane) of the intervals (in the dual plane) to which these end-points belong. This can be done in $O(1)$ time for each event (see Fig. 8).

Lemma 4.1 *The witness set for all the edges $e \in WDG$ can be computed in $O(n^2 \log n + K)$ time using $O(n^2)$ space, for $\beta = \infty$.*

As before, we balance the processing time for intervals and finding the witnesses. We have the following result.

Lemma 4.2 *The set of witnesses for each edge $e \in WDG$ can be computed in $O(n^{3/2} \log n + K)$ time using $O(n)$ space, where K is a count of the total number of times edges in WDG are witnessed for $\beta = \infty$.*

This gives us the following theorem.

Theorem 4.3 *For $\beta \geq 1$, the weighted β-spectrum of P can be constructed in $O(n^{3/2} \log n + K)$ time using $O(n)$ space, where K is a count of the total number of times edges in WDG are witnessed for $\beta = \infty$.*

Similarly, the weighted β-spectrum of P, for $\beta \geq 0$, can be constructed by starting with the complete graph on P as the super graph and determining the witness set for each edge. The following theorem is easily proved.

Theorem 4.4 *The weighted β-spectrum of a set P of n points in the plane, for $\beta \geq 0$, can be constructed in $O(n^2 \log n + K)$ time using $O(n^2)$ space, where K is a count of the total number of times edges that are witnessed for $\beta = \infty$.*

5 Conclusion

In this paper we have proposed efficient algorithms for computing the β-spectrum, $k\beta$-spectrum and weighted β-spectrum in the L_2 metric. Among the open questions are: (i) improve the time complexities of these algorithms (ii) get a tight estimate for K (iii) experiment with using the spectrum as the “signature” of a point set.

References

[Cha86] Bernard Chazelle. Filtering search: A new approach to query-answering. *SIAM J. Computing*, 15(3):703–724, 1986.

[CTL91] M. Chang, C. Tang, and R. Lee. 20-relative neighborhood graphs are hamiltonian. *J. Graph Theory*, 15:543–557, 1991.

[CTL92a] M. Chang, C. Tang, and R. Lee. Solving the euclidean bottleneck biconnected edge subgraph problem by 2-relative neighborhood graph. *Discrete Appl. Math.*, 39:1–12, 1992.

[CTL92b] M. Chang, C. Tang, and R. Lee. Solving the euclidean bottleneck matching problem by k-relative neighborhood graph. *Algorithmica*, 8:177–194, 1992.

[DH73] O. Duda and P. E. Hart. *Pattern classification and scene analysis.* Wiley-Interscience, New York, 1973.

[EGS86] H. Edelsbrunner, L. Guibas, and J. Stolfi. Optimal point location in a monotone subdivision. *SIAM J. Computing*, 15:317–340, 1986.

[EM94] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graphics*, 13(1):43–72, 1994.

[JKY] J. W. Jaromczyk, M. Kowaluk, and F. Yao. An optimal algorithm for constructing β-skeletons in the l_p metric. To be published in SIAM J. Computing.

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

[Lin94] Andrzej Lingas. A linear-time construction of the relative neighborhood graph from the delaunay triangulation. *Comput. Geom. Theory Appl.*, 4:199–208, 1994.

[Mir93] Andy Mirzaian. Minimum weight euclidean matching and weighted relative neighborhood graphs. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes Comput. Sci.*, pages 506–517, 1993.

[RM97] S. V. Rao and Asish Mukhopadhyay. Fast algorithms for computing β-skeletons and their relatives. In *8th Annual International Symposium on Algorithms and Computation*, volume 1350 of *Lecture Notes Comput. Sci.*, pages 374–383. Springer-Verlag, 1997.

[SC90] T. H. Su and R. Ch. Chang. The k-gabriel graphs and their applications. In *Proc. Int. Symp., SIGAL '90*, pages 66–75, 1990.

[SC91] T. H. Su and R. Ch. Chang. Computing the k-relative neighborhood graphs in euclidean plane. *Pattern Recogn.*, 24:231–239, 1991.

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

[Vel92] R. C. Veltkamp. The γ-neighborhood graph. *Comput. Geome. Theory Appl.*, 4:227–246, 1992.