K-set Polygons and Centroid Triangulations

Wael El Oraiby

k-set Polygons and Centroid Triangulations

Wael El Oraiby

A Dissertation submitted for
the degree of Doctor in Computer Science

Université de Haute Alsace
Mulhouse

Laboratoire MIA

August 2009

Contents

Chapter 1

Introduction

A lot of computational and combinatorial geometry problems consist in studying subsets of k elements among n. So, if V is a set of n points in the plane, a k-set of V is a subset of k points of V that are separable from the rest by a straight line.

Since the 1970s, combinatorial geometry is concerned with the number of k-sets in a set of n points. The first results were obtained by Lovász in 1971 in the case where $k = n/2$ (also called halving lines problem) [Lov71]. He showed that every set of n points admits at most $O(n\sqrt{n})$ $n/2$-sets. In 1973, this result was extended to all the values of k by Erdös, Lovász, Simmons and Straus [ELSS73]. They showed that the maximum number of k-sets is in $O(n\sqrt{k})$. They have also shown that there exist sets of points having $\Omega(n \log k)$ k-sets. Since then, better results were found by different authors. The best known results to the day, were given by Dey [Dey98] and Tóth [Tót01]. Dey reduced the upper bound to $O(nk^{\frac{1}{3}})$ while Tóth increased the lower bound to $\Omega(n2^{\Omega(\sqrt{\log k})})$. Reducing the gap between these two bounds remains an important open problem in combinatorial geometry.

A more precise result was obtained by summing the numbers of k-sets of the same set V on different values of k. So, a $(\leq k)$-set is an i-set with $i \leq k$. In 1985, Peck showed that the number of $(\leq k)$-sets is bounded by kn [Pec85]. Note that this value is reached when the points are in convex position. Moreover, Edelsbrunner, Hasan, Seidel and Shen showed that every set of n points admits at least $3\binom{k+1}{2}$ $(\leq k)$-sets when $k \leq n/3$ [EHSS89].

The first contribution of this dissertation from the combinatorial point of view, is a novel summation invariant of the number of k-sets. Contrary to previous work, we fix the value of k and add the number of k-sets over subsets of the set of points V. The subsets are obtained using the following method: Let $\mathcal{V} = (v_1, \ldots, v_n)$ be a sequence of the points of V ordered such that, for all

$i \in \{1, \ldots, n-1\}$, v_{i+1} does not belong to the convex hull of $V_i = \{v_1, \ldots, v_i\}$. Such a sequence is called a convex inclusion chain of V. Every k-set of V_i, for all $i \in \{k+1, \ldots, n\}$, is called k-set of the convex inclusion chain $\mathcal{V}$. We show that the number of k-sets of a convex inclusion chain of V is an invariant of V, that is, it does not depend on the chosen convex inclusion chain $\mathcal{V}$. More precisely, we show that this number is equal to $2kn - n - k^2 + 1 - \sum_{i=1}^{k-1} \gamma^i(V)$, where $\gamma^i(V)$ is the number of i-sets of V.

The previous result is obtained using the so called k-set polygon of V. This polygon is one of the numerous objects that are linked to k-sets and that contributed to their reputation. The k-set polygon of a point set V in the plane (called k-set polytope in arbitrary dimension) was introduced by Edelsbrunner, Valter and Welzl in 1997 [EVW97]. It consists of the convex hull of the centroids of all subsets of k points of V. Andrzejak and Fukuda showed that the vertices of the k-set polygon of V are precisely the centroids of the k-sets of V [AF99]. Hence, counting the k-sets of V comes to count the number of vertices of the k-set polygon of V. More precisely in our case, the number of k-sets of a convex inclusion chain $\mathcal{V} = (v_1, \ldots, v_n)$ of V is equal to the total number of distinct vertices of the k-set polygons of $V_{k+1}, \ldots, V_n$, that is, the total number of distinct vertices found by an algorithm that successively builds the k-set polygons of $V_{k+1}, \ldots, V_n$.

k-set construction problem was studied extensively in computational geometry. In 1986, Edelsbrunner and Welzl gave an algorithm that allows the construction of the k-sets of a set of n points in $O(m \log^2 n)$ time, where m is the size of the output. This result was enhanced (for smaller values of k) by Cole, Sharir, and Yap [CSY87]. After sorting the set of points, they determine every k-set in $O(\log^2 k)$ time per k-set and so reaching the total complexity of $O(n \log n + m \log^2 k)$, where m is the size of the output. This algorithm is currently the best algorithm known. It should be noted that the $\log^2 k$ factor comes from using the dynamic convex hull data structure by Overmars and von Leeuwen [OvL81]. Replacing this data structure by the data structure of Brodal and Jacob [BJ02], this factor can be then reduced to $\log k$ amortized time. Everett, Robert and van Kreveld showed that the $(\leq k)$-sets can be found in $O(n \log n + nk)$ time, which is optimal in the worst case [ERvK93].

In this dissertation, we propose an on-line algorithm that constructs the k-sets of particular convex inclusion chains: The ones that form simple polygonal lines. This comes to give an on-line algorithm to construct the k-set polygon of a simple polygonal line which is such that every point of this line is outside the convex hull of the points preceding it. This algorithm generalizes, in some way, the on-line algorithm of Melkman, that builds the convex hull of a simple polygonal line in linear time [Mel87]. The complexity of our algorithm is in $O(c \log^2 k)$ where c

is the total number of constructed k-sets. The complexity per k-set, is then the same as the one in the algorithm of Cole, Sharir and Yap.

Since the k-set polygon of a set of points is a convex hull, we can then adapt other convex hull construction algorithms to determine the k-sets. It is important to note that the algorithm of Cole, Sharir and Yap is inspired by the gift-wrapping algorithm of Chand and Kapur [CK70], also called the Jarvis march [Jar73] that builds the convex hull of n points in $O(mn)$ time, where m is the size of the constructed convex hull. Another classical algorithm for constructing the convex hull is the divide and conquer algorithm by Preparata and Hong [PH77]. This algorithm has a complexity of $O(n \log n)$ and presents the advantage that it can be extended to the third dimension, with the same complexity.

We adapt here this algorithm to find the k-sets in the plane. The algorithm we propose has a complexity in $O(n \log n + c \log^2 k \log(n/k))$, where c is the maximum number of k-sets of a set of n points. We will see that the $\log(n/k)$ factor comes from an overestimation done while computing the complexity.

Among the other notions closely related to k-sets in an arbitrary dimension, we can cite the k-levels in the hyperplane arrangements in the dual space [Ede87, CGL85], the k-hulls [CSY87], the order-k Voronoi diagrams [Aur91, SS98], the order-k Delaunay diagram [SS06], half-space range searching [CP86], orthogonal L_1-line fitting [YKII88], corner cuts (notably used in the field of computer vision) [OS99], ...

Given a set V of points in the plane, the order-k Voronoi diagram of V is a partition of the plane whose every region is associated to a subset T of k points of V. More precisely, the order-k Voronoi region associated to T is the subset of points in the plane that are closer to each of the elements of T than to any other element of $V \setminus T$. A first relation between the k-sets and the order-k Voronoi diagrams is that the k-sets of V are the subsets of V associated to the unbounded order-k Voronoi regions of V. Moreover, every order-k Voronoi region in the plane (resp. in dimension d) corresponds to a k-set in the third dimension (resp. in dimension $d+1$) [Ede87].

In 1982, Lee proposed an iterative algorithm that builds the order-k Voronoi diagram in the plane from the order-$(k-1)$ Voronoi diagram [Lee82]. Starting with the order-1 Voronoi diagram, the algorithm builds the order-k Voronoi diagram in $O(k^2 n \log n)$ time. In the same article, Lee showed that the number of order-k Voronoi regions of a set V of n points in the plane, no four of them being co-circular, is equal to $2kn - n - k^2 + 1 - \sum_{i=1}^{k-1} \gamma^i(V)$ where $\gamma^i(V)$ is the number of i-sets of V. Now, this number is the same as the number of k-sets of a convex inclusion chain of V, that we have found in this dissertation. In a surprising way, this gives us a new relation between k-sets and the order-k Voronoi diagram. In the rest of this dissertation, we try to understand this result.

In 1991, Schmitt and Spehner showed that the order-k Voronoi diagram admits a dual whose vertices are the centroids of the k-subsets of V associated to the order-k Voronoi regions [SS91]. This dual is called the order-k Delaunay diagram of V, or order-k Delaunay triangulation when V does not contain four co-circular points. Aurenhammer and Schwarzkopf showed that this dual is actually a projection of a three dimensional convex polyhedral surface [AS92]. More precisely, the order-k Delaunay triangulation is a projection of the lower part of a k-set polytope of dimension 3 [SS06]. In the plane, the order-k Delaunay triangulation decomposes the k-set polygon of V into two types of triangles: So called territory triangles and domain triangles. The order-1 Delaunay triangulation is the classical Delaunay triangulation; it contains only territory triangles. These observations allow the adaptation of Lee's iterative algorithm to the construction of order-k Delaunay triangulation [SS91]. For $k \geq 2$, the method consists, at the first step, in deducing the order-$(k+1)$ domain triangles from the order-k territory triangles. The order-$(k+1)$ territory triangles are obtained in a second step by computing the constrained order-1 Delaunay triangulation of the $(k+1)$-set polygon of V deprived of the order-$(k+1)$ domain triangles. Being inside "nearly-convex" polygons, the constrained Delaunay triangulations can be built by the linear algorithm of Aggarwal, Guibas, Saxe and Shor [AGSS89]. The order-k Delaunay triangulation is then built iteratively in $O(n \log n + k^2(n-k))$ time [Sch95].

In 2004, Neamtu showed how the order-k Delaunay triangulation can be used to generate multivariate splines [Nea04]. Liu and Snoeyink, noticed though that the types of splines generated in such a way are limited [LS07]. They proposed to define a larger triangulation family called order-k centroid triangulations, that contains the order-k Delaunay triangulation. The order-k centroid triangulations of a set of points V are also triangulations of the k-set polygon of V whose vertices are the centroids of subsets of k points of V and whose triangles are territory and domain triangles. A centroid triangulation is defined in a constructive way using a generalization of the iterative order-k Delaunay construction algorithm. The essential difference with this algorithm is that the second step constructs arbitrary constrained triangulations (and not necessarily Delaunay ones). Moreover, the algorithm does not start from an order-1 Delaunay triangulation, but from any triangulation of V. However, Liu and Snoeyink were not successful in proving the validity of their algorithm for $k > 3$. Experimental results indicates that the algorithm also works for the cases where $k \geq 4$. Nevertheless, the only triangulations we actually have the proof that they belong to centroid triangulation family for all k, are the order-k Delaunay triangulations.

In this dissertation, we show that, for every convex inclusion chain $\mathcal{V}$ of V, the centroids of the k-sets of $\mathcal{V}$ are the vertices of an order-k centroid triangulation, enlarging by the occasion the family of centroid triangulations. The result is actually a first step toward the understanding of the fact that the number of k-sets of a convex inclusion chain of V is equal to the number of order-k Voronoi regions of V. Indeed, the centroids of the k-sets of a convex inclusion chain of V, on the one hand, and the centroids of the subsets of k points associated to the order-k Voronoi regions of V, on the other hand, are the vertices of an order-k centroid triangulation of V. To complete the argument, one needs to prove that all the order-k centroid triangulations of V, as defined by Liu and Snoeyink, have the same number of vertices.

We give here one sufficient condition so that the property is verified: It suffices that every maximal edge-connected set of domain triangles in a centroid triangulation is convex.

In this case, every order-k centroid triangulation has a size in $O(k(n-k))$ and is built by the algorithm of Liu and Snoeyink in at least $O(n \log n + k^2(n-k))$ time. We show here that a particular order-k centroid triangulation can be constructed in $O(n \log n + k(n-k)\log^2 k)$ time.

This dissertation is composed of five chapters in addition to the introduction and the conclusion.

In the second chapter, we recall some important results on the convex hulls, the k-set polygons, the order-k Voronoi diagrams, the order-k Delaunay triangulations, and the centroid triangulations.

In the third chapter, we introduce the convex inclusion chains and show that their number of k-sets, is an invariant of the considered point set.

In the fourth chapter, we give an on-line construction algorithm of the k-sets of a convex inclusion chain that forms a simple polygonal line.

In the fifth chapter, we propose a divide and conquer algorithm to find the k-sets of a set of points in the plane.

In the sixth chapter, we show that the centroids of the k-sets of a convex inclusion chain are the vertices of an order-k centroid triangulation. We give also an algorithm to construct a particular centroid triangulation. Finally, we give a sufficient condition so that all the order-k centroid triangulations of the same set of points have the same number of vertices.

Chapter 2

k-sets and related objects

2.1 Introduction

This chapter assembles some generalizations on the k-element subsets taken from a finite set of points V in the plane. It also contains properties that will be used in the next chapters.

To every subset T of k points of V in the plane, we associate its centroid $g(T)$. The convex hull of the set of these centroids is called the k-set polygon of V. We prove that the vertices of this polygon are the centroids of the sets of points that are separable from the others by a straight line. We start by recalling some properties on the convex sets to apply them later on the k-set polygons.

By selecting subsets of k points of V, we can also define a triangulation of the k-set polygon of V.

A typical example is the order-k Delaunay triangulation of V that is dual to the order-k Voronoi diagram of V. This triangulation is obtained by selecting the subsets of k points of V that are separable from the rest of the points of V by circles. An algorithm for building the order-k Delaunay triangulation is also given.

This algorithm is then generalized to other types of triangulations of the k-set polygon whose vertices are also centroids of subsets of k points of V. These types of triangulations are called the order-k centroid triangulations and will be the subject of the last chapter in this dissertation.

2.2 Convex sets and convex hulls

We recall here some definitions and properties of the convex sets used in the sections that follow. In addition, we should note that the convex hull of a set of points V in the plane is the k-set polygon with $k = 1$.

The following symbols and conventions are used through the whole dissertation.

Given a finite point set V in the plane, we suppose that no three points of V are collinear.

For every subset $\mathcal{P}$ of the plane, we denote by $\mathring{\mathcal{P}}$ the relative interior of $\mathcal{P}$, by $\overline{\mathcal{P}}$ the closure of $\mathcal{P}$, and by $\delta(\mathcal{P}) = \overline{\mathcal{P}} \setminus \mathring{\mathcal{P}}$ the boundary of $\mathcal{P}$.

Given two points s and t in the plane, st denotes the closed oriented line segment going from s to t and (st) denotes the oriented straight line spanned by st.

Given an oriented straight line Δ, Δ^+ (resp. Δ^-) is the closed half plane on the left (resp. right) side of Δ.

Definition 2.1 (of Convex Set)**.** *Let $\mathcal{C}$ be a set of points in the Euclidean space. $\mathcal{C}$ is said to be convex if, for all points s and t in $\mathcal{C}$ and for every real number μ such that $0 \leq \mu \leq 1$, the point $(1-\mu)s + \mu t$ is in $\mathcal{C}$ (see Figure 2.1).*

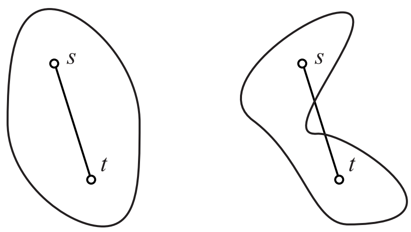

Figure 2.1: An example of a convex set (left) and a non-convex set (right)

Property 2.2. *(i) The intersection of a non-empty family of convex sets is convex if it is not empty.*

(ii) For every non empty subset V of the Euclidean space, there exists a smallest convex subset of the space that contains V.

Proof. (i) Let $(C_i)_{i \in I}$ be a non-empty family of convex subsets such that $C = \cap_{i \in I} C_i \neq \emptyset$. For all s and t of C and for every $i \in I$, s and t belong to C_i. Since C_i is convex, for every $\mu \in [0,1]$, $(1-\mu)s+\mu t \in C_i$. It results that $(1-\mu)s+\mu t \in C$ and that C is convex.

(ii) Since the Euclidean space is convex and contains the set V, the family of all convex sets that contain V is not empty and its intersection C contains V. From (i), C is convex. C is then the smallest convex subset of the Euclidean space containing V. □

Definition 2.3 (of Convex Hull)**.** *For every subset $V \neq \emptyset$ of the Euclidean space E, the smallest convex subset of E which contains V is called the convex hull of V and is denoted by $conv(V)$ (see Figure 2.2).*

By Property 2.2, the convex hull of V in the Euclidean space is the intersection of all convex sets containing V.

Then, it is not hard to prove:

Property 2.4. *For every set* $V = \{v_1, v_2, ...v_n\}$,

$$conv(V) = \left\{ \sum_{i=1}^{n} \lambda_i v_i \ for \ \lambda_i \geq 0 \ and \ \sum_{i=1}^{n} \lambda_i = 1 \right\}.$$

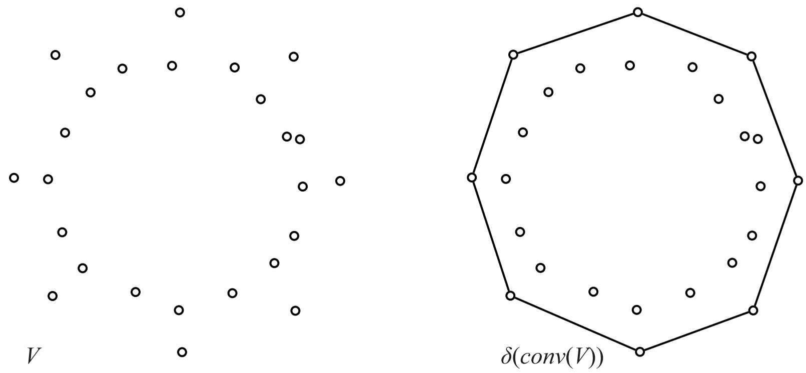

Figure 2.2: The point set V and its convex hull

Definition 2.5 (of Extreme Point)**.** *A point v of a convex set V is said to be an extreme point if $conv(V \setminus \{v\})$ does not contain v. Every extreme point v of a convex set V is also called a vertex of $\delta(conv(V))$ and, for the sake of simplicity, we will say that this vertex is also a vertex of $conv(V)$.*

One can also prove that:

Property 2.6. *Let v be a point of a finite set V. The following properties are equivalent (see Figure 2.3):*

(i) v is a vertex of $conv(V)$;

(ii) There exists a straight line Δ that separates v from $V \setminus \{v\}$;

(iii) There exists a straight line Δ' that passes through v and such that $\Delta' \cap conv(V) = \{v\}$;

Property 2.7. *Let s and t be two points of V. The line segment st is an edge of $conv(V)$ if, and only if, V is included in one of the half planes on the left or on the right of the straight line st.*

Remark 2.8. *Let $p_1, p_1, ...p_m$ be the vertices of $\delta(conv(V))$ in the counter clockwise direction. It results from Property 2.7 that, for every edge p_ip_{i+1} of $conv(V)$, $V \subset (p_ip_{i+1})^+$.*

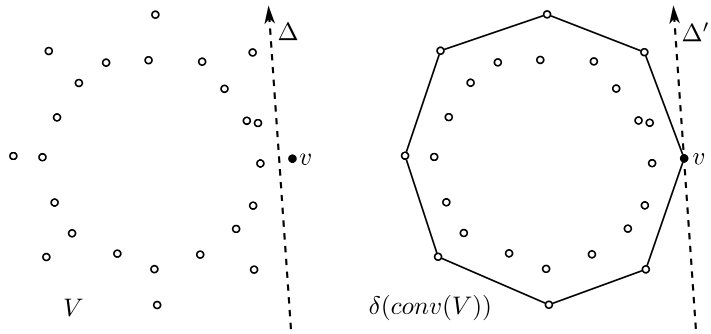

Figure 2.3: The point v (black) can be separated from $V \setminus v$ (white points) by an oriented straight line Δ, such that $v \in \mathring{\Delta}^-$. If Δ' is parallel to Δ passing through v, oriented as Δ, $conv(V) \subset \Delta'^+$.

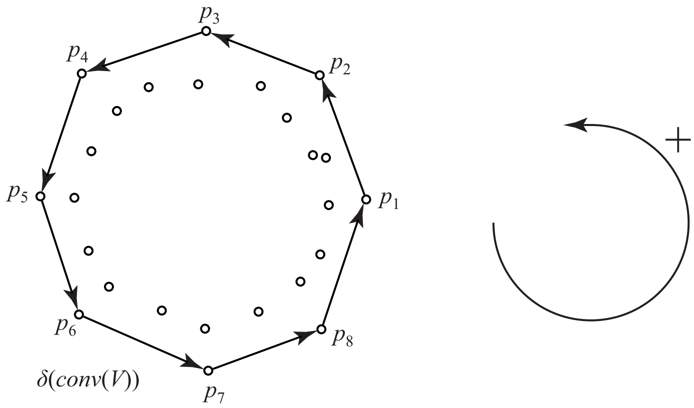

Figure 2.4: $conv(V)$, its vertices $p_1, p_2, ..., p_8$, and its oriented edges $p_1p_2, p_2p_3, ..., p_8p_1$.

2.3 k-set polygon

Let V be a set of n points in the plane and k an integer such that $k \in \{1, \ldots, n\}$. By associating to each subset T of k points of V its centroid $g(T)$ we find a set of points whose convex hull is called the k-set polygon of V noted by $g^k(V)$ (see Figure 2.5). We characterize here, the vertices and the edges of a k-set polygon. We show that a subset T of k points of V defines a vertex of the k-set polygon if and only if T is separable from $V \setminus T$ by a straight line. In this case, the set T is called a k-set of V.

In the particular case where $k = n$, V is the unique k-set of V and $g^k(V)$ is reduced to the center of gravity of V.

For the rest of this chapter we suppose that $k < n$ and we study the relations between k-sets and k-set polygons in this case. For this, we need the following two technical lemmas:

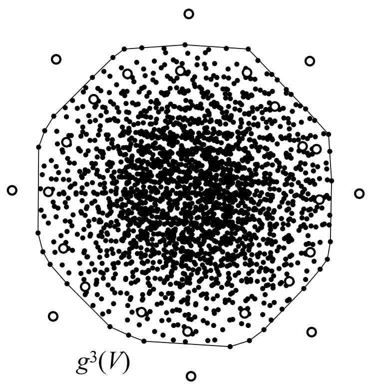

Figure 2.5: The centers of gravity for all 3-element subsets of a set V, and the 3-set polygon of V, $g^3(V)$

Lemma 2.9. *Let T be a non-empty subset of k elements of V, U a non-empty subset of T, and Δ an oriented straight line such that $U \subset \Delta^+$. Let Δ' be the straight line parallel to Δ, with the same orientation as Δ, and that passes through $g(T)$.*

(i) For every subset R of $V \setminus T$ with same cardinality as U and included in Δ^-, the centroid of $T' = (T \setminus U) \cup R$ belongs to Δ'^-.

(ii) Moreover, if at least one point of U belongs to $\mathring{\Delta}^+$ or one point of R belongs to $\mathring{\Delta}^-$, then $g(T') \in \mathring{\Delta}'^-$.

Proof. (i) Let π be a straight line orthogonal to Δ oriented from Δ^+ to Δ^-. Consider now the abscissae of the points of the plane on π: The abscissa of $g(T)$ on π is the average of the abscissae of the points of T on π (see Figure 2.6). Since the abscissae of the points of R on π are greater than or equal to the abscissae of the points of U on π, the average of the abscissae of the points of $T' = (T \setminus U) \cup R$ is greater than or equal to the abscissa of $g(T)$. Thus $g(T')$ belongs to Δ'^-.

(ii) Moreover, if at least one point of U belongs to $\mathring{\Delta}^+$ or one point of R belongs to $\mathring{\Delta}^-$, the abscissa of at least one point of R is strictly greater than the abscissa of one point of U. The abscissa of $g(T')$ is then strictly greater than the abscissa of $g(T)$ and $g(T')$ belongs to $\mathring{\Delta}'^-$. □

Lemma 2.10. *A subset T of k points of V is strictly separable from the other points of V by a straight line Δ if and only if $g(T)$ is strictly separable from the centers of gravity of the other subsets of k points of V by a straight line Δ' parallel to Δ. Moreover, if Δ and Δ' have the same orientation and if $T \subset \mathring{\Delta}^-$ then $g(T) \in \mathring{\Delta}'^-$.*

Proof. (i) Let T be a subset of k points of V such that there exists an oriented straight line Δ such that $T \subset \mathring{\Delta}^-$ and $V \setminus T \subset \mathring{\Delta}^+$. Let Δ'' be an oriented straight

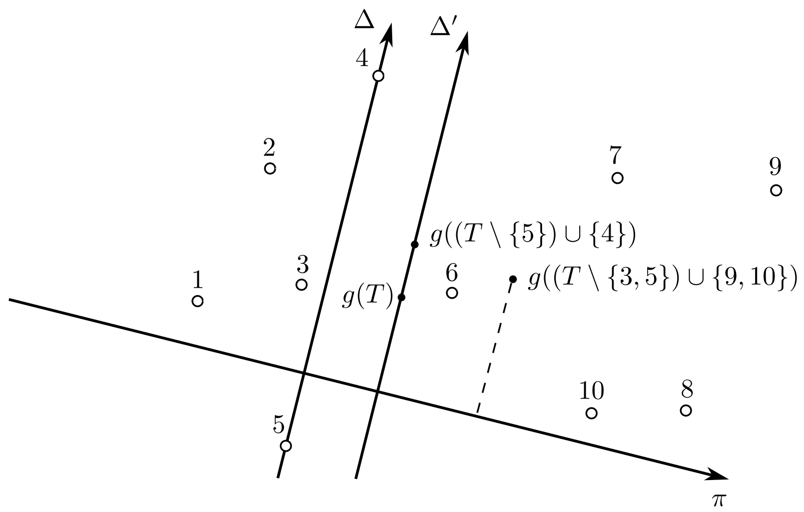

Figure 2.6: illustration of the proof for lemma 2.9 with $T = \{1, 2, 3, 5, 6, 7, 8\}$

line parallel to Δ, with the same direction as Δ, and that passes through $g(T)$. Every subset T' of k points of V distinct from T admits at least one point that belongs to $\mathring{\Delta}^+$. From lemma 2.9, it results that $g(T')$ belongs to $\mathring{\Delta}''^+$. Hence, $g^k(V) \cap \Delta'' = \{g(T)\}$ and, from Property 2.6, $g(T)$ is separable from all the other centroids of sets of k points of V by a straight line Δ' parallel to Δ'' and Δ. Moreover, if Δ' is oriented as Δ, $g(T) \subset \mathring{\Delta}'^-$.

(ii) Conversely, let T be a subset of k points of V such that there exists an oriented straight line Δ' that strictly separates $g(T)$ from the other centers of gravity of the k-point subsets of V and such that $g(T) \subset \mathring{\Delta}'^-$.

Let Δ'' be a straight line parallel to Δ' that passes through $g(T)$ and oriented in the same direction as Δ'. From Property 2.6, $g^k(V) \cap \Delta'' = \{g(T)\}$ and $g^k(V) \subset \Delta''^-$. Let Δ''' be the straight line parallel to Δ', that passes through a point t of T, oriented in the same direction as Δ' and such that $T \subset \Delta'''^-$. There exists then a straight line Δ parallel to Δ''', oriented in the same direction as Δ''' such that $\Delta \subset \mathring{\Delta}'''^+$ and such that $\Delta^- \cap \mathring{\Delta}'''^+$ does not contain any point of V. If $(V \setminus T) \cap \mathring{\Delta}^- \neq \emptyset$, there exists a point s in $(V \setminus T) \cap \Delta^-$ and this point belongs also to Δ'''^-. Hence, from Lemma 2.9, $g((T \setminus \{t\}) \cup \{s\})$ belongs to Δ'''^- which contradicts the previous results. It results that T is strictly separable from $V \setminus T$ by Δ and belongs to $\mathring{\Delta}^-$. □

The following result is a direct consequence of this lemma (see Figure 2.7):

Proposition 2.11. *T is a k-set of V if and only if its centroid $g(T)$ is a vertex of $g^k(V)$.*

Moreover if two k-sets T and T' are distinct then $g(T)$ and $g(T')$ are distinct vertices.

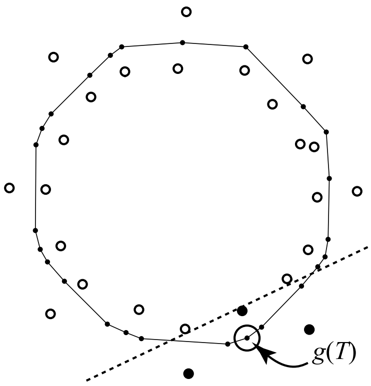

Figure 2.7: The 3-set T (bold black points) and its centroid $g(T)$

The edges of $g^k(V)$ can be characterized using the following proposition:

Proposition 2.12. *T and T' are two k-sets of V such that $g(T)g(T')$ is a counter clockwise oriented edge of $g^k(V)$, if and only if there exist two points s and t of V and a subset P of $k-1$ points of V such that $T = P \cup \{s\}$, $T' = P \cup \{t\}$, and $V \cap (\mathring{st})^- = P$.*

Proof. (i) Let s and t be two points of V such that $|V \cap (\mathring{st})^-| = k - 1$ and let $P = V \cap (\mathring{st})^-$ (see Figure 2.8). Let Δ' be the straight line parallel to (st), with the same direction as (st), and that passes through $g(P \cup \{s\})$. Δ' is then the image of (st) by an homothety of center $g(P)$ and of ratio $1/k$. It results that $g(P \cup \{t\})$ also belongs to Δ' and that the segments st and $g(P \cup \{s\})g(P \cup \{t\})$ have the same direction.

Let U be a subset of k points of V distinct from $P \cup \{s\}$ and from $P \cup \{t\}$. If $U \not\subset P \cup \{s, t\}$, U contains at least one point of $(\mathring{st})^+$ and otherwise, at least one point of P (*i.e* of $V \cap (\mathring{st})^-$) does not belong to U. In both cases, from Lemma 2.9, $g(U)$ belongs to $\mathring{\Delta}'^+$. It results that $g^k(V) \subset (g(P \cup \{s\})\mathring{}g(P \cup \{t\}))^+$ and therefore, $g(P \cup \{s\})g(P \cup \{t\})$ is a counter clockwise oriented edge of $g^k(V)$.

(ii) Now, let $g(T)g(T')$ be a counter clockwise oriented edge of $g^k(V)$ and Δ' the oriented straight line spanned by $g(T)g(T')$ and oriented from $g(T)$ to $g(T')$. Let Δ be the oriented straight line parallel to Δ', with the same direction as Δ', that passes through a point of $T \cup T'$, and such that $T \cup T' \subset \Delta^-$. If there exists a subset U of k points of V in $\mathring{\Delta}^-$, $g(U)$ belongs to $\mathring{\Delta}'^-$ from lemma 2.9. This is impossible because $g^k(V) \subset \Delta'^+$ by construction. Hence $|(T \cup T') \cap \mathring{\Delta}^-| < k$. Now, since $T \neq T'$, we have $|T \cup T'| > k$. Since no three points of V are collinear, it results that $P = T \cap T'$ contains exactly $k-1$ points. Hence, Δ passes through exactly two points s and t of $T \cup T'$. By choosing s and t such that (st) is oriented as Δ', it results from (i) that $T = P \cup \{s\}$ and $T' = P \cup \{t\}$.

□

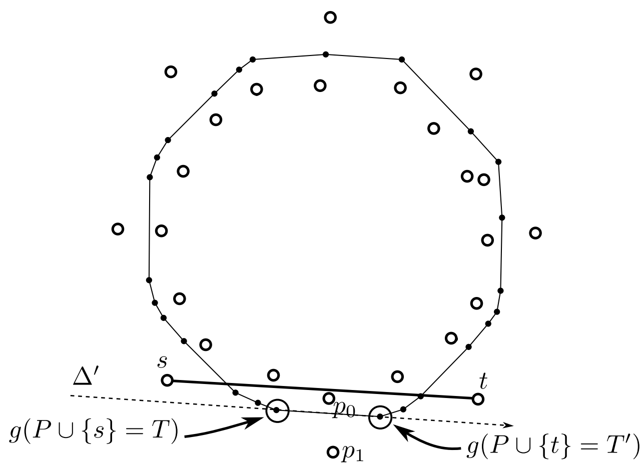

Figure 2.8: Illustration of the proof of Proposition 2.12 with $k = 3$ and $P = \{p_0, p_1\}$

The closed oriented edge $g(P \cup \{s\})g(P \cup \{t\})$ of $g^k(V)$ is denoted hereafter by $e_P(s,t)$.

Note that, in the particular case where $k = 1$, $g^k(V)$ is the convex hull of V and its edges are of the form $e_\emptyset(s,t)$. When V is reduced to two points s and t, $g^1(V)$ admits exactly two oriented edges $e_\emptyset(s,t) = st$ and $e_\emptyset(t,s) = ts$.

Corollary 2.13. *If $e_{P_i}(s_i,t_i)$ is an edge of $g^k(V)$ and $e_{P_{i+1}}(s_{i+1},t_{i+1})$ is its successor on $\delta(g^k(V))$, then $P_i \cup \{t_i\} = P_{i+1} \cup \{s_{i+1}\}$, $s_it_i \cap s_{i+1}t_{i+1} \neq \emptyset$, and $s_{i+1} \in (s_it_i)^-$.*

Proof. Since $g(P_i \cup \{t_i\}) = g(P_{i+1} \cup \{s_{i+1}\})$, from proposition 2.11, $P_i \cup \{t_i\} = P_{i+1} \cup \{s_{i+1}\}$ and, from proposition 2.12, $P_i \cup \{t_i\} \subset (s_it_i)^-$ (see Figure 2.9). Thus $s_{i+1} \in P_i \cup \{t_i\} \subset (s_it_i)^-$. Moreover, since $t_{i+1} \notin P_{i+1} \cup \{s_{i+1}\} = P_i \cup \{t_i\}$, $t_{i+1} \in (s_it_i)^+$. Thus $(s_it_i) \cap s_{i+1}t_{i+1} \neq \emptyset$. In the same way $s_it_i \cap (s_{i+1}t_{i+1}) \neq \emptyset$ and thus $s_it_i \cap s_{i+1}t_{i+1} \neq \emptyset$. $\square$

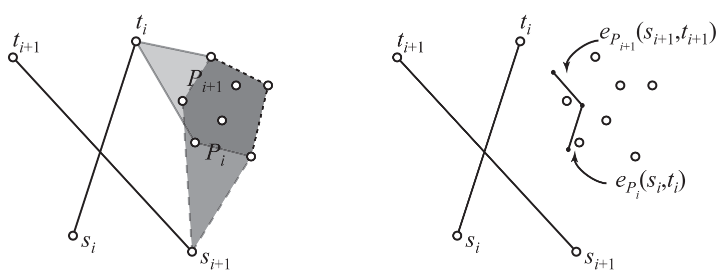

Figure 2.9: Illustration of the proof of corollary 2.13 using 9-set polygon edges

2.4 Order-k Voronoi diagram

We are still working on the set V of n points in the plane and an integer $k \in \{1, \ldots, n-1\}$. For every subset T of k points of V, let $\mathcal{R}(T)$ be the set of points in the plane that are strictly closer to each of the points of T than to any point of $V \setminus T$. If $\mathcal{R}(T)$ is not empty, $\mathcal{R}(T)$ is called an order-k Voronoi region of V, and more precisely, the order-k Voronoi region of T (see Figure 2.10).

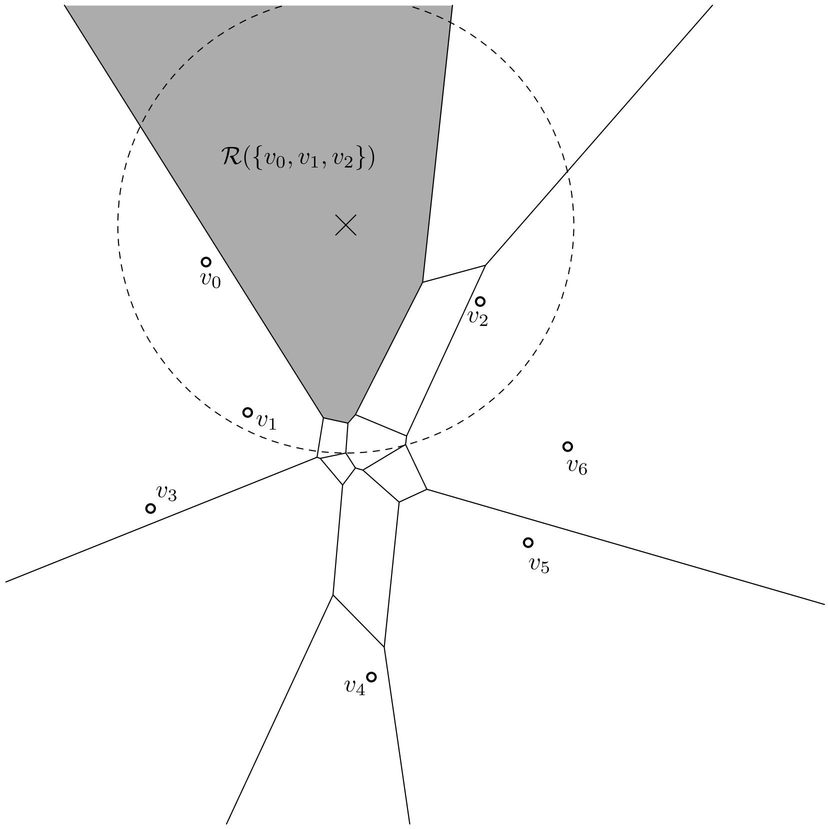

Figure 2.10: The order-3 Voronoi region $\mathcal{R}(\{v_0, v_1, v_2\})$ (grey color). All the points in this region are closer to the points $\{v_0, v_1, v_2\}$ than to any point of $\{v_3, \ldots, v_6\}$.

The set of the order-k Voronoi regions of V, their edges and their vertices form a partition of the plane called the order-k Voronoi diagram of V.

Note that $\mathcal{R}(T)$ is not empty if and only if T is strictly separable from $V \setminus T$ by a circle. $\mathcal{R}(T)$ is then the set of centers for such circles.

Note also that a region $\mathcal{R}(T)$ is unbounded if and only if some of these circles have centers that tend toward infinity, that is, if and only if T is a k-set of V.

The order-k Voronoi edges can also be characterized by empty circles in the following way:

Property 2.14. *$\mathcal{R}(T)$ and $\mathcal{R}(T')$ are two order-k Voronoi regions of V, sharing a common edge if and only if there exist two distinct points s and t of V, a subset P of $k-1$ points of $V \setminus \{s,t\}$, and a circle σ such that $T = P \cup \{s\}$, $T' = P \cup \{t\}$, s and t are on σ, P is inside σ, and all the other points of $V \setminus (P \cup \{s,t\})$ are outside σ.*

The common edge of $\mathcal{R}(T)$ and $\mathcal{R}(T')$ is then the set of centers of such circles σ (see Figure 2.11).

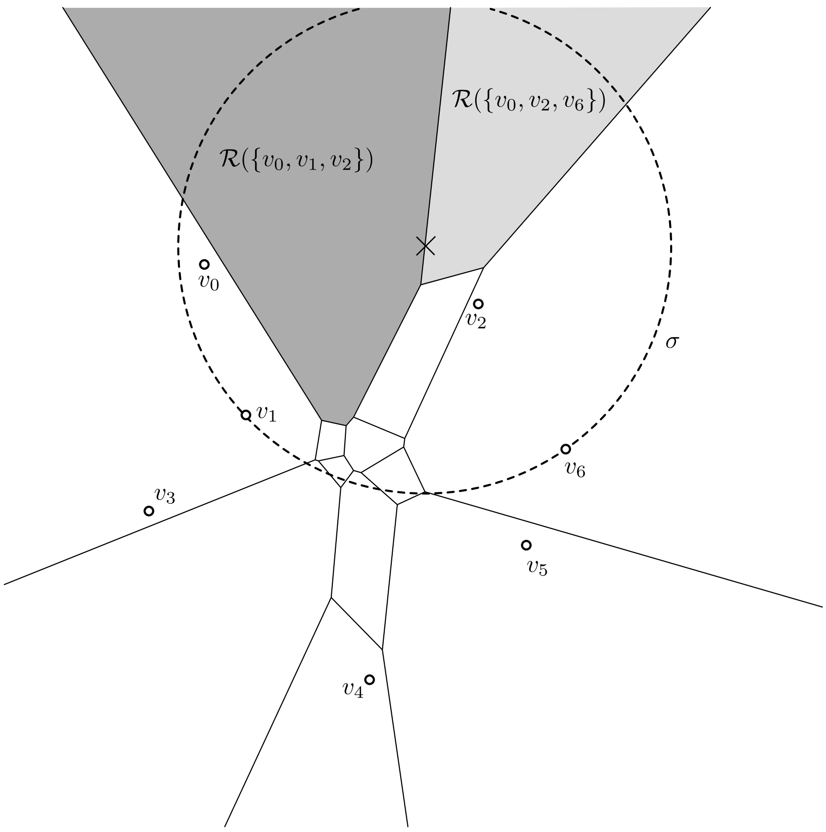

Figure 2.11: Two adjacent order-3 Voronoi regions $\mathcal{R}(\{v_0, v_1, v_2\})$ and $\mathcal{R}(\{v_0, v_2, v_6\})$. Their common edge is the set of centers of the circles σ that contain $P = \{v_0, v_2\}$ and pass through $s = v_1$ and $t = v_6$.

Remark 2.15. *Note that such an order-k Voronoi edge is unbounded if and only if some of the circles σ have centers that tend toward infinity, that is, if and only if the straight line (st) strictly separates P from $V \setminus P$. In this case, the line segment that links $g(P \cup \{s\})$ to $g(P \cup \{t\})$ is an edge of the k-set polygon of V from Proposition 2.12.*

In the same way, for the vertices of the order-k Voronoi diagram we have the following characterization:

Property 2.16. *Every order-k Voronoi vertex of V is the center of a circle that passes through three points r, s, t of V, that contains a set P of either $k-1$ or $k-2$ points of V inside, and such that all the other points of V are outside the circle.*

Conversely, every point in the plane that is the center of such a circle is an order-k Voronoi vertex of V.

In addition, if $|P| = k-1$, the vertex is common to the three order-k Voronoi regions $\mathcal{R}(P \cup \{r\})$, $\mathcal{R}(P \cup \{s\})$, $\mathcal{R}(P \cup \{t\})$ and if $|P| = k-2$, the vertex is common to the three order-k Voronoi regions $\mathcal{R}(P \cup \{r,s\})$, $\mathcal{R}(P \cup \{s,t\})$, $\mathcal{R}(P \cup \{r,t\})$ (see Figure 2.12).

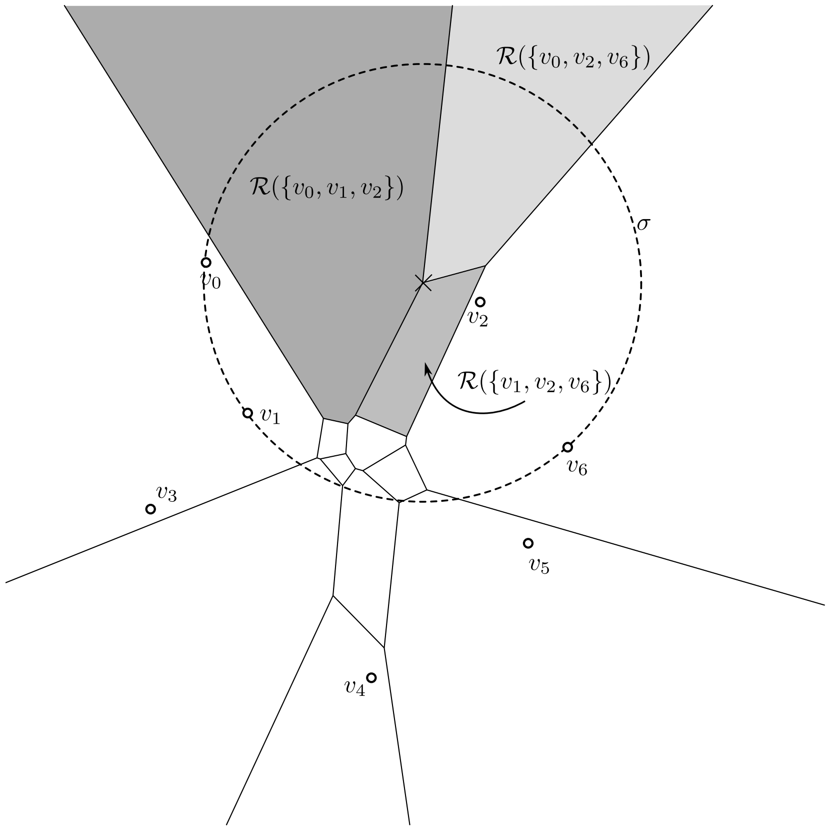

Figure 2.12: The order-3 Voronoi vertex adjacent to the three regions $\mathcal{R}(P \cup \{r,s\})$, $\mathcal{R}(P \cup \{s,t\})$, $\mathcal{R}(P \cup \{r,t\})$ with $P = \{v_2\}$, $r = v_0$, $s = v_1$, and $t = v_6$. The circle σ centred at this vertex contains $P = \{v_2\}$ and passes through r, s, t.

Remark 2.17. *Note that when $|P| = k-1$, such a Voronoi vertex is also an order-$(k+1)$ Voronoi vertex common to the three order-$(k+1)$ Voronoi regions $\mathcal{R}(P \cup \{r,s\})$, $\mathcal{R}(P \cup \{s,t\})$, $\mathcal{R}(P \cup \{r,t\})$, and when $|P| = k-2$, it is an order-$(k-1)$ Voronoi vertex common to the three order-$(k-1)$ Voronoi regions $\mathcal{R}(P \cup \{r\})$, $\mathcal{R}(P \cup \{s\})$, $\mathcal{R}(P \cup \{t\})$.*

By convention, when $|V| = k$, the order-k Voronoi diagram of V is composed of the unique region $\mathcal{R}(V)$ that is the whole plane.

In the particular case where $V = \{s, t\}$ and $k = 1$, the order-k Voronoi diagram of V is composed of the two regions $\mathcal{R}(s)$ and $\mathcal{R}(t)$ that are two half planes delimited by the bisector of line segment st. This bisector is the unique edge of the diagram.

In the general case where $|V| > 2$ and $|V| > k$, the order-k Voronoi diagram of V admits at least one vertex (we have supposed that no three points of V are collinear). Moreover, the edges of this diagram are line segments and half lines.

Lee [Lee82] proved that the number of regions of this diagram is equal to $2kn - n - k^2 + 1 - \sum_{i=1}^{k-1} \gamma^i(V)$, where $n = |V|$ and $\gamma^i(V)$ is the number of i-sets of V.

2.5 Order-k Delaunay triangulation

We show that for every subset T that defines an order-k Voronoi region, the centroid $g(T)$ is a vertex of a dual diagram called the order-k Delaunay triangulation of V. This diagram is a particular triangulation of the k-set polygon. We will give an algorithm that constructs the order-k Delaunay triangulation by iterations over k.

2.5.1 Definitions

- For every order-k Voronoi region $\mathcal{R}(T)$, the centroid $g(T)$ of T is called an order-k Delaunay vertex (see Figure 2.13).

- For every order-k Voronoi edge common to two regions $\mathcal{R}(P \cup \{s\})$ and $\mathcal{R}(P \cup \{t\})$, the line segment $g(P \cup \{s\})g(P \cup \{t\})$ is called an order-k Delaunay edge (see Figure 2.13).

- For every order-k Voronoi vertex common to three regions $\mathcal{R}(P \cup \{r\})$, $\mathcal{R}(P \cup \{s\})$, $\mathcal{R}(P \cup \{t\})$, the triangle $g(P \cup \{r\})g(P \cup \{s\})g(P \cup \{t\})$ is called an order-k Delaunay triangle. We will also call this triangle an order-k Delaunay territory triangle (see Figure 2.14).

- For every order-k Voronoi vertex common to the regions $\mathcal{R}(P \cup \{r, s\})$, $\mathcal{R}(P \cup \{s, t\})$, $\mathcal{R}(P \cup \{r, t\})$, the triangle $g(P \cup \{r, s\})g(P \cup \{s, t\})g(P \cup \{r, t\})$ is also called an order-k Delaunay triangle. We will also call this triangle an order-k Delaunay domain triangle (see Figure 2.15).

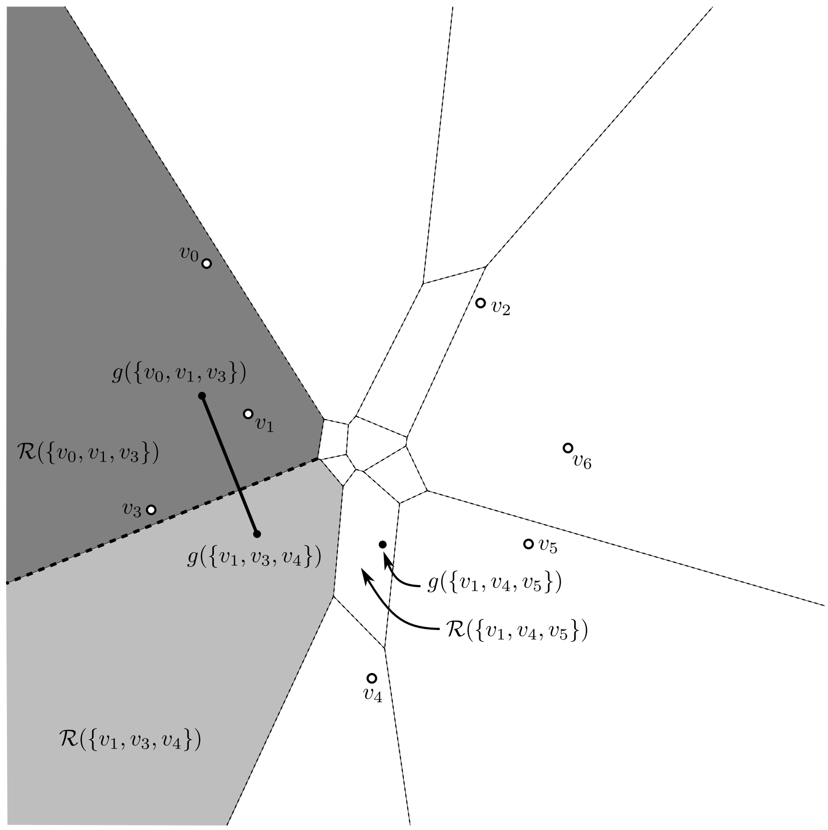

Figure 2.13: The order-3 Delaunay vertex $g(\{v_1, v_4, v_5\})$ is dual to the order-3 Voronoi region $\mathcal{R}(\{v_1, v_4, v_5\})$. The order-3 Delaunay edge $g(\{v_1, v_3\} \cup \{v_0\})g(\{v_1, v_3\} \cup \{v_4\})$ is dual to the edge shared by the order-3 Voronoi regions $\mathcal{R}(\{v_1, v_3\} \cup \{v_0\})$ and $\mathcal{R}(\{v_1, v_3\} \cup \{v_4\})$. Note that the two edges are orthogonal.

Proposition 2.18. *The order-k Delaunay vertices, the open order-k Delaunay edges, and the open order-k Delaunay triangles are pairwise disjoint.*

Proof. From the properties of the order-k Voronoi regions, edges and vertices, every order-k Delaunay vertex, edge or triangle ψ of V can be characterized by a couple (P, Q) of subsets of V such that:

- $P \cap Q = \emptyset$,
- there exists a closed disk ω such that $\mathring{\omega} \cap V = P$ and $\delta(\omega) \cap V = Q$,
- $|P| = k$ and $Q = \emptyset$, if ψ is a vertex,
- $|P| = k - 1$ and $|Q| = 2$, if ψ is an edge,
- $k - 2 \leq |P| \leq k - 1$ and $|Q| = 3$, if ψ is a triangle.

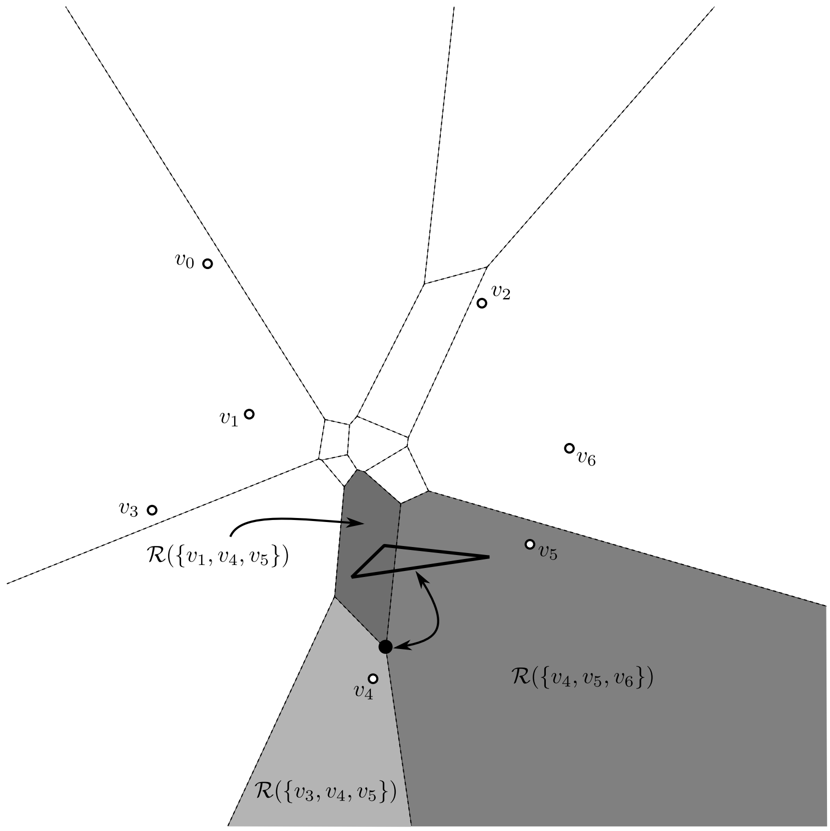

Figure 2.14: The order-3 Delaunay territory triangle is dual to the order-3 Voronoi vertex common to the regions $\mathcal{R}(\{v_4, v_5\} \cup \{v_1\})$, $\mathcal{R}(\{v_4, v_5\} \cup \{v_3\})$, and $\mathcal{R}(\{v_4, v_5\} \cup \{v_6\})$

In addition, the vertices of ψ are the centroids of the subsets of k points of $P \cup Q$ that contain P.

Let ψ' be an order-k Delaunay vertex, edge or triangle of V distinct from ψ, (P', Q') the couple of subsets of V that characterizes ψ', and ω' a closed disk such that $\mathring{\omega}' \cap V = P'$ and $\delta(\omega') \cap V = Q'$. Since $\psi' \neq \psi$, we have $(P', Q') \neq (P, Q)$ and, hence, $\omega' \neq \omega$.

(i) Obviously, when ω and ω' are disjoint, ψ and ψ' are disjoint as well since $\psi \subseteq conv(P \cup Q) \subset \omega$ and $\psi' \subseteq conv(P' \cup Q') \subset \omega'$.

(ii) Let us now prove by contradiction that none of the disks ω and ω' contains the other. Within a permutation of ω and ω', let us suppose that $\omega' \subset \omega$.

If $\delta(\omega) \cap \delta(\omega') \cap V = \emptyset$, $P' \cup Q' \subseteq P$. Then, from the cardinality constraints on P, Q, P', and Q', we necessarily have $Q = Q' = \emptyset$ and $P = P'$. This is impossible since $(P, Q) \neq (P', Q')$.

In the same way, if $\delta(\omega) \cap \delta(\omega') \cap V = \{q\}$ then $P' \cup (Q' \setminus \{q\}) \subseteq P$. Now, since Q and Q' are not empty, we have $|P' \cup Q'| > k$ and $|P| < k$, which is impossible.

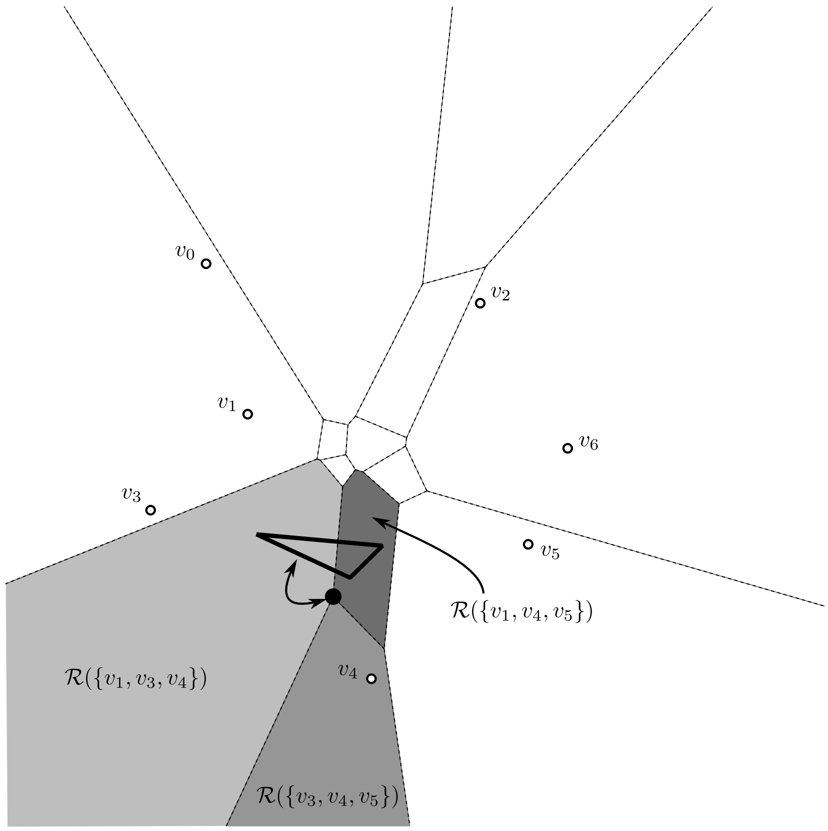

Figure 2.15: The order-3 Delaunay domain triangle is dual to the order-3 Voronoi vertex common to the regions $\mathcal{R}(\{v_4\} \cup \{v_1, v_3\})$, $\mathcal{R}(\{v_4\} \cup \{v_3, v_5\})$, and $\mathcal{R}(\{v_4\} \cup \{v_1, v_5\})$

(iii) The remaining case to study is the case where $\delta(\omega)$ intersects $\delta(\omega')$ in one or two points and where none of ω and ω' is inside the other. Let Δ be the oriented straight line such that $\Delta \cap \delta(\omega) = \Delta \cap \delta(\omega') = \delta(\omega) \cap \delta(\omega')$ and $\omega \cap \Delta^- \subset \omega'$ (*i.e.* $\omega' \cap \Delta^+ \subset \omega$). Since $\omega \cap V = P \cup Q$ and $\omega' \cap V = P' \cup Q'$, we then have $(P \cup Q) \setminus P' \subset \Delta^+$ and $(P' \cup Q') \setminus P \subset \Delta^-$ (see Figure 2.16).

Let us show by contradiction that $((P \cup Q) \setminus P') \cup ((P' \cup Q') \setminus P) \not\subset \Delta$. Indeed, since $\Delta \cap \mathring{\omega} = \Delta \cap \mathring{\omega}'$, $P = \mathring{\omega} \cap V$, and $P' = \mathring{\omega}' \cap V$, we have $\Delta \cap \mathring{\omega} \cap V = \Delta \cap \mathring{\omega}' \cap V \subseteq P \cap P'$. If we suppose that $((P \cup Q) \setminus P') \cup ((P' \cup Q') \setminus P) \subset \Delta$ then $(P \setminus P') \cup (P' \setminus P) \subset \Delta$ and this implies that $(P \setminus P') \cup (P' \setminus P) \subseteq P \cap P'$ and therefore, $P = P' = \emptyset$. $((P \cup Q) \setminus P') \cup ((P' \cup Q') \setminus P) \subset \Delta$ becomes then $Q \cup Q' \subset \Delta$. Now, $Q = \delta(\omega) \cap V$, $Q' = \delta(\omega') \cap V$ and $\delta(\omega) \cap \Delta = \delta(\omega') \cap \Delta$. It results that $Q = Q'$, which is impossible from the hypothesis $(P, Q) \neq (P', Q')$.

We can then suppose, within a permutation of (P, Q) and (P', Q'), that there exists a point p of $(P \cup Q) \setminus P'$ that does not belong to Δ. This point then necessarily belongs to Δ^+. Let Δ' be a straight line parallel to Δ, with the same direction as Δ, and that passes through the vertex $g(T')$ of ψ' and such that

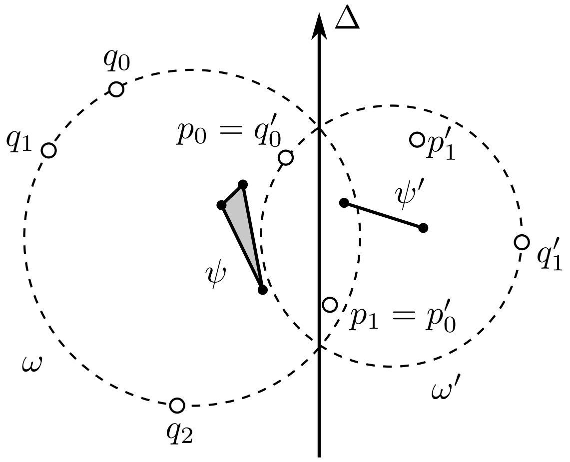

Figure 2.16: An order-3 Delaunay territory triangle ψ characterized by the couple (P, Q) such that $P = \{p_0, p_1\}$ and $Q = \{q_0, q_1, q_2\}$ that is disjoint from an order-3 Delaunay edge ψ' characterized by the couple (P', Q') such that $P' = \{p'_0, p'_1\}$ and $Q' = \{q'_0, q'_1\}$.

$\psi' \subset \Delta'^-$. For every vertex $g(T)$ of ψ, since $T \subseteq P \cup Q$ and since $P' \subseteq T'$, we have $T \setminus T' \subseteq (P \cup Q) \setminus P' \subset \Delta^+$. Likewise, $T' \subseteq P' \cup Q'$ and $P \subseteq T$ implies that $T' \setminus T \subseteq (P' \cup Q') \setminus P \subset \Delta^-$. It results from Lemma 2.9 that $g(T) \in \Delta'^+$. Moreover, for at least one of the vertices $g(T)$ of ψ, $p \in T \setminus T'$ and, therefore, $g(T) \in \mathring{\Delta}^+$, from Lemma 2.9. It results that $\mathring{\psi} \subset \mathring{\Delta}^+$ and, since $\psi' \subset \Delta^-$, $\mathring{\psi} \cap \mathring{\psi}' = \emptyset$. $\square$

Theorem 2.19. *The order-k Delaunay vertices, edges and triangles of V form a triangulation of the k-set polygon $g^k(V)$ of V.*

Proof. (i) First, let us consider the general case where $|V| > 2$ and $|V| > k$. In this case, from Section 2.4, the order-k Voronoi diagram of V admits at least one vertex. Hence, there exists at least one Delaunay triangle. Moreover, by definition, the order-k Delaunay vertices and edges are the vertices and edges of the order-k Delaunay triangles. Since these vertices are centroids of subsets of k points of V, all the triangles are inside the k-set polygon of V. In addition, from proposition 2.18, the open order-k Delaunay triangles, the open order-k Delaunay edges and the order-k Delaunay vertices are pairwise disjoint.

It remains to prove that the set of the order-k Delaunay triangle covers $g^k(V)$. From Remark 2.15, every order-k Delaunay edge c that is not an edge of $g^k(V)$ is dual to a bounded edge c' of the order-k Voronoi diagram. By duality, c is then an edge common to two order-k Delaunay triangles that are dual to both ends of c'. This proves that the triangles of the order-k Delaunay triangulation cover $g^k(V)$.

(ii) By definition, in the particular case where $|V| = k$, the order-k Voronoi diagram of V is composed of the unique region $\mathcal{R}(V)$. $g(V)$ is then the unique order-k Delaunay vertex and is equal to the k-set polygon of V.

If $V = \{s, t\}$ and $k = 1$, the order-k Voronoi diagram of V is composed of the two regions $\mathcal{R}(s)$ and $\mathcal{R}(t)$ and of their unique common edge. By duality, s and t are then the two unique order-k Delaunay vertices and the line segment that links them is the unique order-k Delaunay edge. In this case the k-set polygon of V is equal to the convex hull of V and is also reduced to this segment. □

The obtained triangulation is called the order-k Delaunay triangulation (see Figure 2.17). Note that in the particular case where $k = 1$, we recognize the definition of the (classical) Delaunay triangulation whose triangles have their vertices in V and are inscribed in circles containing $k - 1 = 0$ points inside.

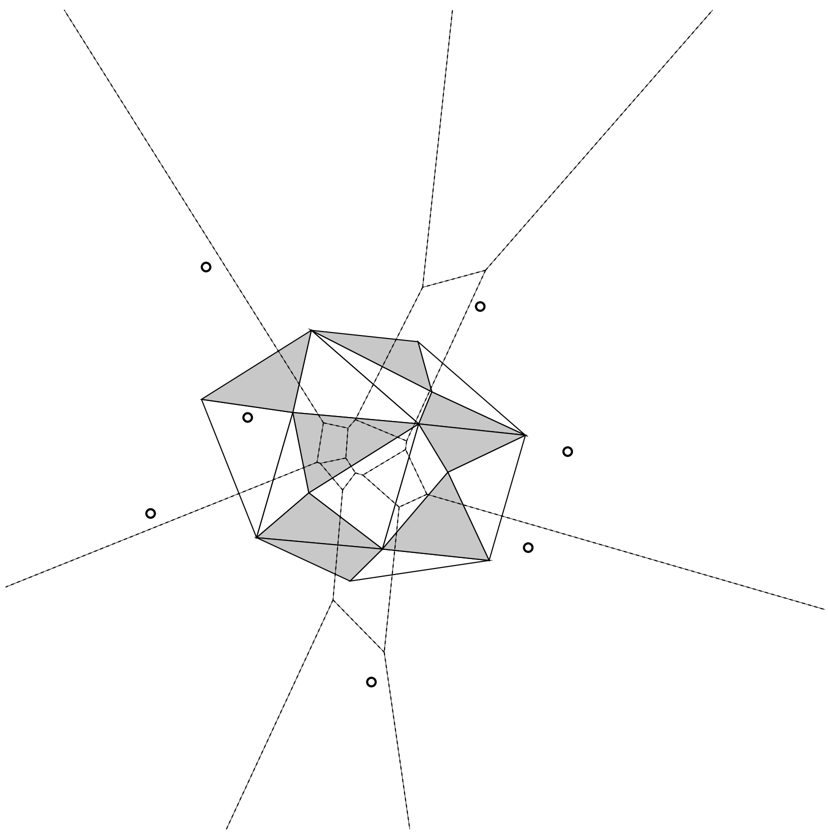

Figure 2.17: The order-3 Delaunay triangulation partitions the 3-set polygon and is dual to the order-3 Voronoi diagram. Gray triangles are domain triangles and white triangles are territory triangles.

2.5.2 Construction

We will show now how the order-k Delaunay triangulation can be constructed from the order-$(k-1)$ Delaunay triangulation when $k \geq 2$. Since the order-k Delaunay triangles are dual to the order-k Voronoi vertices, it results from Remark 2.17 that the order-k Delaunay domain triangles can be constructed from the order-$(k-1)$ Delaunay territory triangles. Hence, $g(P \cup \{r, s\})g(P \cup \{s, t\})g(P \cup \{t, r\})$ is an order-k domain triangle if and only if $g(P \cup \{r\})g(P \cup \{s\})g(P \cup \{t\})$ is an order-$(k-1)$ territory triangle.

Moreover, the set of order-k Delaunay vertices is determined in this way as shown by the following lemma:

Lemma 2.20. *If $k \geq 2$, every order-k Delaunay vertex is a vertex of an order-k Delaunay domain triangle.*

Proof. If $g(T)$ is an order-k Delaunay vertex, $\mathcal{R}(T)$ is an order-k Voronoi region and, by definition, there exists a circle that strictly separates T from $V \setminus T$. By reducing the circle radius while keeping the center, we can find a circle σ that passes through a point s of T and such that the rest of the points of T are on σ or inside σ. Then, there exists a circle σ' tangent to σ in s, that passes through a second point t of T and such that all the other points of T are on σ' or inside σ'. Clearly, all the points of $V \setminus T$ are outside σ'. Since $|V| > |T|$, we can suppose within a permutation of s and t, that $(\mathring{st})^+$ contains at least one point of $V \setminus T$. Hence, there exists necessarily a circle σ'' that passes through s and t (*i.e.* whose center is on the bisector of st), that passes also, either through a point r of $(V \setminus T) \cap (\mathring{st})^+$ (Case 1) or through a point t' of $T \cap (\mathring{st})^-$ (Case 2), and such that all the other points of T are on or inside σ'' and all the points of $V \setminus T$ are on or outside σ''.

Case 1 (Figure 2.18): In this case, from the definition of the order-k Delaunay triangles, $g((T \setminus \{s, t\}) \cup \{r, s\})$, $g((T \setminus \{s, t\}) \cup \{s, t\}) = g(T)$ and $g((T \setminus \{s, t\}) \cup \{r, t\})$ are the vertices of an order-k Delaunay domain triangle.

Case 2 (Figure 2.19): In the second case, within a permutation of (st') and (tt'), $(st')^+$ contains a point of $V \setminus T$ (since $(\mathring{st})^+ \subset (\mathring{st'})^+ \cup (\mathring{tt'})^+$). We can then restart the previous process by replacing (st) with (st'). Since $(\mathring{st'})^- \cap T \subset (\mathring{st})^- \cap T$, after at most $k-2$ iterations of this process, we necessarily end up finding a circle that passes through two points s and t of T, through one point r of $V \setminus T$, and such that all the other points of $T \setminus \{s, t\}$ are inside the circle and the points of $V \setminus (T \cup \{r\})$ are outside the circle. Thus, $g(T)$ is a vertex of a domain triangle, from Case 1. $\square$

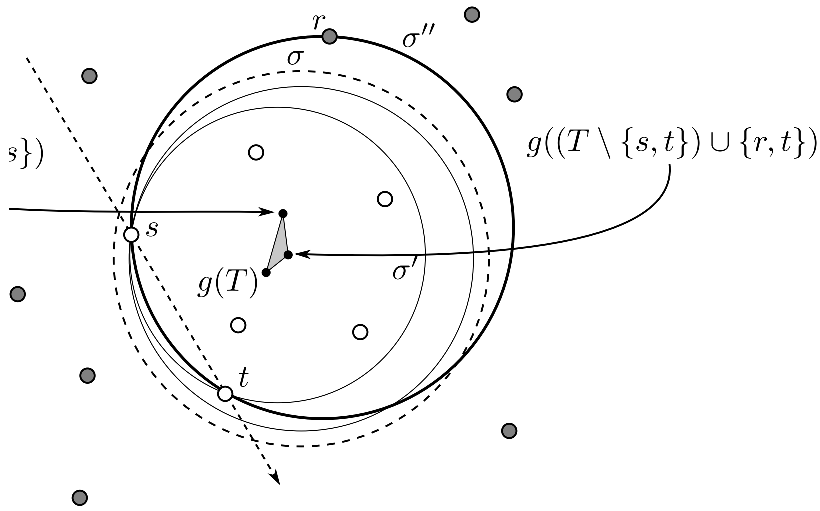

Figure 2.18: Illustration of the proof of Lemma 2.20: Case 1 with $k = 6$ (the points of T are in white)

From this lemma, the vertices of the order-k Delaunay territory triangles are vertices of the order-k Delaunay domain triangles. It results that, if τ is the set of order-k Delaunay domain triangles then, the order-k Delaunay territory triangles form a constrained triangulation of $\overline{g^k(V) \setminus \tau}$.

Recall that a constrained triangulation of $\overline{g^k(V) \setminus \tau}$ is a partition of $\overline{g^k(V) \setminus \tau}$ in triangles such that the vertices of $\overline{g^k(V) \setminus \tau}$ are the vertices of the partition and that each edge of $\overline{g^k(V) \setminus \tau}$ is also an edge of the partition.

If for some subset P of $k-1$ points of V, the set formed by the order-k Delaunay edges that are of the form $g(P \cup \{s\})g(P \cup \{t\})$ and by the order-k Delaunay territory triangles that are of the form $g(P \cup \{r\})g(P \cup \{s\})g(P \cup \{t\})$ is not empty, then this set is called the territory of P.

Proposition 2.21. *Let $\mathcal{C}$ be the closure of a connected component of $g^k(V) \setminus \tau$. The order-k Delaunay triangles inside $\mathcal{C}$ belong to the same territory and form a constrained (order-1) Delaunay triangulation of $\mathcal{C}$.*

Proof. (i) By definition, the edges of a triangle $g(P \cup \{s\})g(P \cup \{r\})g(P \cup \{t\})$ of the territory of P, belong also to the territory of P. Since every order-k Delaunay edge belongs to one and only one territory, two territory triangles cannot have a common edge unless they belong to the same territory. Since, by construction, all the order-k Delaunay triangles belonging to $\mathcal{C}$ are territory triangles, it results that these triangles belong to the same territory.

(ii) From (i), there exists a subset P of $k-1$ points of V and a set $S = \{s_1, \ldots, s_m\}$ of points of $V \setminus P$ such that $g(P \cup \{s_1\}), \ldots, g(P \cup \{s_m\})$ are the

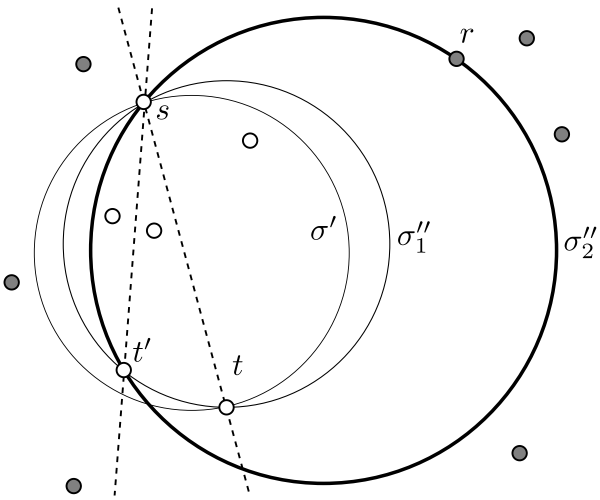

Figure 2.19: Case 2 of the proof of Lemma 2.20. σ'' is found in two iterations of the process.

vertices of $\mathcal{C}$. For every order-k Delaunay triangle $g(P\cup\{s_i\})g(P\cup\{s_j\})g(P\cup\{s_l\})$ the points of $S \setminus \{s_i, s_j, s_l\}$ are outside the circle that passes through s_i, s_j, s_l. It results that the triangle $s_i s_j s_l$ is an (order-1) Delaunay triangle of S. By an homothety of center $g(P)$ and of ratio $1/k$, $g(P \cup \{s_i\})g(P \cup \{s_j\})g(P \cup \{s_l\})$ is then an (order-1) Delaunay triangle of $\{g(P \cup \{s_1\}), \ldots, g(P \cup \{s_m\})\}$. If, in addition, this triangle is inside $\mathcal{C}$, then it is a triangle of the constrained (order-1) Delaunay triangulation of $\mathcal{C}$. □

It results from this proposition that, when the set τ of order-k Delaunay domain triangles is known, the territory triangles can be obtained by computing the constrained (order-1) Delaunay triangulation of $\overline{g^k(V) \setminus \tau}$. Hence, the algorithm that constructs the order-k Delaunay triangulation from the order-$(k-1)$ Delaunay triangulation:

```
function build_order_k_Delaunay
{
  1. foreach (order-(k-1) Delaunay territory triangle
       g(P ∪ {r})g(P ∪ {s})g(P ∪ {t}))
  {
    compute the triangle g(P ∪ {r,s})g(P ∪ {s,t})g(P ∪ {r,t});
  }
```

```
  2. compute the constrained (order-1) Delaunay triangulation
     of g^k(V) \ τ (where τ is the set of the triangles built in
     loop 1);
}
```

In Figures 2.20 and 2.21, we can see the two steps of the algorithm in the cases $k = 2$ and $k = 3$.

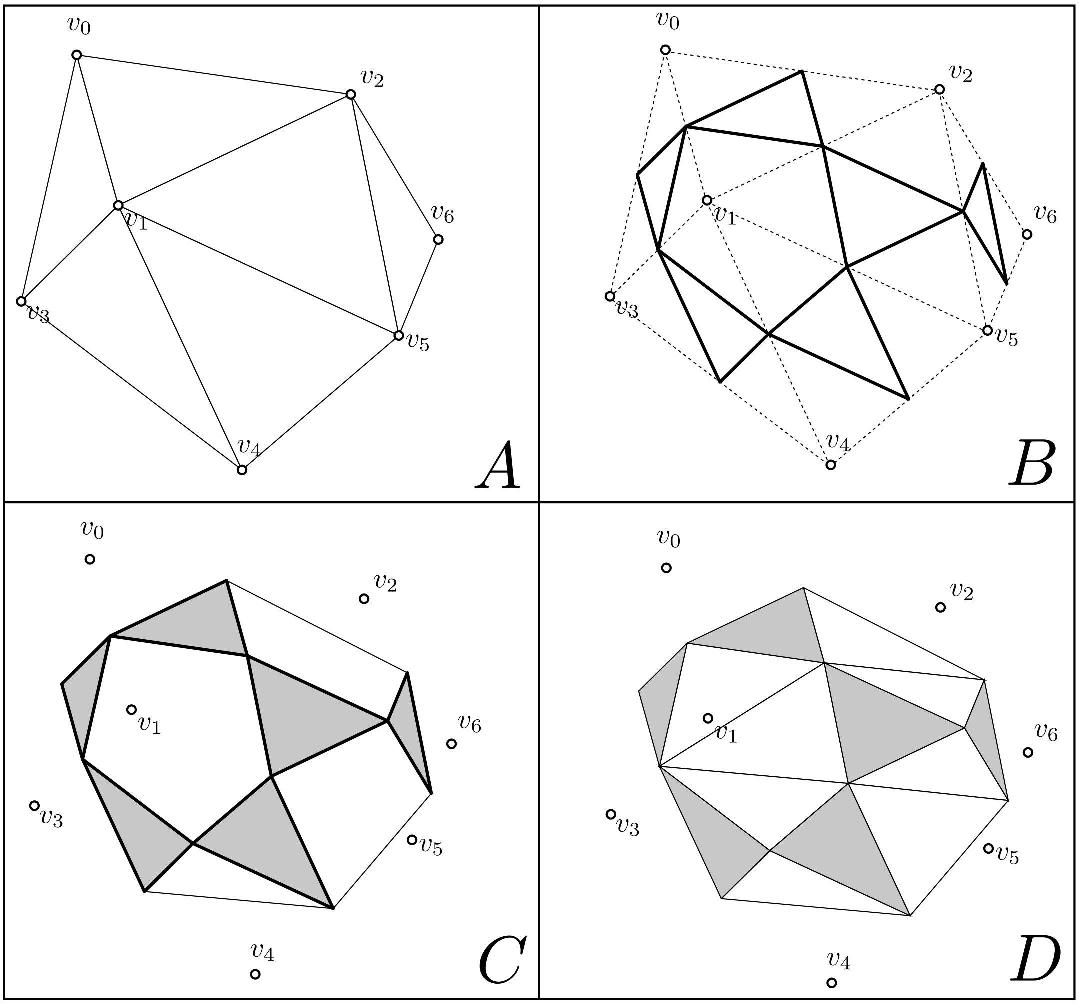

Figure 2.20: Illustration of the order-2 Delaunay triangulation construction: For each order-1 Delaunay triangle we build its corresponding order-2 Delaunay domain triangle (A and B). We later compute the constrained Delaunay triangulation of the 2-set polygon without the domain triangles and we get the order-2 Delaunay triangulation (C and D).

Before giving the complexity of this algorithm, recall that the (order-1) Delaunay triangulation inside a convex polygon, can be computed in linear time with the algorithm of Aggarwal *et al.* [AGSS89]. Using the results of Lee [Lee82]

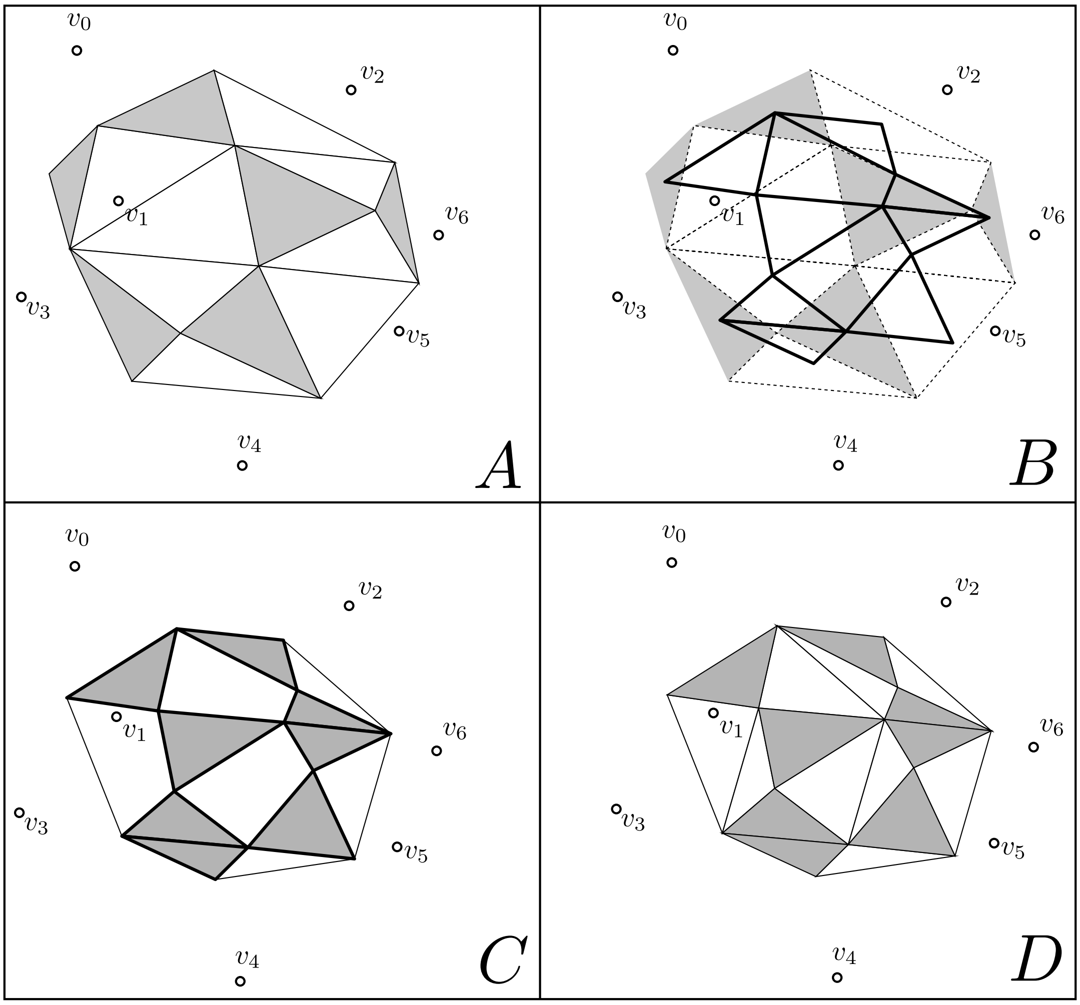

Figure 2.21: Illustration of the order-3 Delaunay triangulation construction: For each order-2 Delaunay triangle we build its corresponding order-3 Delaunay domain triangle (A and B). We later compute the constrained Delaunay triangulation of the 3-set polygon without the domain triangles and we get the order-3 Delaunay triangulation (C and D).

on the order-k Voronoi diagram, Aggarwal *et al.* have also proved that each territory, even if it is not convex and not simple, has enough interesting properties that makes it possible for their algorithm to be used. It results then from proposition 2.21 that the above algorithm constructs the order-k Delaunay triangulation from the order-$(k-1)$ Delaunay triangulation in time linear with its size. Now, as we have seen in the previous section, the order-k Voronoi diagram of V admits $O(k(n-k))$ regions and, by duality, the size of the order-k Delaunay triangulation of V is also in $O(k(n-k))$.

Thus, given a set of points V, its order-k Delaunay triangulation can be constructed iteratively in $O(n \log n + k^2(n-k))$ time, by first constructing the order-1 Delaunay triangulation of V in $O(n \log n)$ time and then applying $k-1$ times the previous algorithm.

2.6 Order-k centroid triangulations

We have seen in Section 2.3, that the vertices of the k-set polygon of V are the centroids of the k-sets of V and that these centroids are pairwise disjoint. Moreover, the edges of $g^k(V)$ are of the form $g(T)g(T')$ with $|T \cap T'| = k-1$.

In the previous section, we have seen that the order-k Delaunay triangulation of V is a triangulation of $g^k(V)$ whose vertices are the centroids of the k-point subsets of V that are separable from the others by circles. Here again, these centroids are pairwise disjoint and the order-k Delaunay triangulation edges are of the form $g(T)g(T')$ with $|T \cap T'| = k-1$.

More generally, we call k-neighbor triangulation of V every triangulation $\mathcal{T}$ of $g^k(V)$ such that:

- there exists a set $\mathcal{P}$ of k-point subsets of V such that every vertex of $\mathcal{T}$ is the centroid of a unique element of $\mathcal{P}$.
- every edge of $\mathcal{T}$ is of the form $g(T)g(T')$ with $\{T, T'\} \subseteq \mathcal{P}$ and $|T \cap T'| = k-1$.

From this definition, if V admits different k-point subsets that have the same centroid, then at most one of these subsets is in $\mathcal{P}$.

Afterwards, when we say that $g(T)$ is a vertex of a k-neighbor triangulation, this will mean that T belongs to $\mathcal{P}$.

Moreover, if the centroid of every element of $\mathcal{P}$ is a vertex of $\mathcal{T}$, we say that $\mathcal{P}$ determines the vertices of $\mathcal{T}$.

Property 2.22. *If $\mathcal{T}$ is a k-neighbor triangulation of V then:*
(i) $\mathcal{T}$ admits only two types of triangles:

- *triangles of the form $g(P \cup \{r\})g(P \cup \{s\})g(P \cup \{t\})$, where P is a $(k-1)$-point subset of V and where r, s, t are three distinct points of $V \setminus P$,*
- *triangles of the form $g(P \cup \{r, s\})g(P \cup \{s, t\})g(P \cup \{r, t\})$, where P is a $(k-2)$-point subset of V and where r, s, t are three distinct points of $V \setminus P$.*

(ii) Moreover, when $k = 1$ (resp. $k = n-1$), all the triangles of $\mathcal{T}$ are of the first (resp. second) type.

Proof. (i) Let $g(T)g(T')g(T'')$ be a triangle of $\mathcal{T}$. Since $|T \cap T'| = |T' \cap T''| = |T \cap T''| = k-1$, we have necessarily $k-2 \leq |T \cap T' \cap T''| \leq k-1$ and $k+1 \leq |T \cup T' \cup T''| \leq k+2$. Thus, there exist a subset P of V and three distinct points r, s, t of $V \setminus P$ such that:

- either $T = P \cup \{r\}$, $T' = P \cup \{s\}$, $T'' = P \cup \{t\}$,
- or $T = P \cup \{r, s\}$, $T' = P \cup \{s, t\}$, $T'' = P \cup \{r, t\}$.

(ii) When $k = 1$, every triangle of $\mathcal{T}$ is of the form rst (with $\{r, s, t\} \subseteq V$) and is then of the first type (with $P = \emptyset$).

When $k = n-1$, every triangle of $\mathcal{T}$ is of the form $g(V \setminus \{r\})g(V \setminus \{s\})g(V \setminus \{t\})$ (with $\{r, s, t\} \subseteq V$) and is then of the second type (with $P = V \setminus \{r, s, t\}$). $\square$

As in the case of the order-k Delaunay triangulation, we call these two types of triangles territory triangles and domain triangles respectively.

It is important to note that Property 2.22 is wrong if the set $\mathcal{P}$ that determines the vertices of $\mathcal{T}$ contains two elements that have the same centroid (see Figure 2.22).

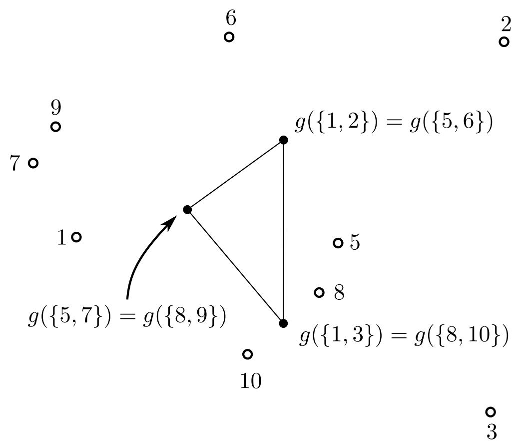

Figure 2.22: The three edges $g(\{1,3\})g(\{1,2\})$, $g(\{5,6\})g(\{5,7\})$, and $g(\{8,9\})g(\{8,10\})$ of the triangle are of the form $g(T)g(T')$ with $|T \cap T'| = 1$. However, the triangle is neither an order-2 territory triangle nor an order-2 domain triangle.

In the case of the order-k Delaunay triangulations, the bijection between territory triangles at the order $k-1$ and domain triangles at the order k, allowed us to give an algorithm that constructs the order-k Delaunay triangulation from the order-$(k-1)$ Delaunay triangulation. We can now extend this algorithm to the k-neighbor triangulations.

Let us note first that every triangulation of V is a 1-neighbor triangulation of V. So, we also call the triangulations of V order-1 (centroid) triangulations of V.

Now, let $k > 1$ and let $\mathcal{T}$ be a $(k-1)$-neighbor triangulation of V. Consider the following generalization of the algorithm `build_order_k_Delaunay`:

```
function build_order_k_triangulation
{
  1. foreach (territory triangle g(P∪{r})g(P∪{s})g(P∪{t}) of T)
  {
    compute the triangle g(P∪{r,s})g(P∪{s,t})g(P∪{r,t});
  }

  2. compute a constrained triangulation of overline(g^k(V) \ τ) (where τ is
       the set of the triangles built in loop 1);
}
```

Let us study the result of the application of this algorithm on an order-1 triangulation $\mathcal{T}^1$ of V (see Figure 2.23).

From Property 2.22, every triangle of $\mathcal{T}^1$ is a territory triangle. τ is then the set of triangles obtained by constructing, for each triangle rst of $\mathcal{T}^1$, a triangle whose vertices are the midpoints $g(\{r,s\})$, $g(\{s,t\})$, $g(\{r,t\})$ of the edges of rst. These triangles are pairwise disjoint except at their vertices. Note also that every vertex of the 2-set polygon $g^2(V)$ is a vertex of such a triangle. Indeed, if $g(\{s,t\})$ is a vertex of $g^2(V)$, then from proposition 2.11, $\{s,t\}$ is strictly separable from $V \setminus \{s,t\}$ by a straight line. Therefore, the line segment st is an edge of every (order-1) triangulation of V. It results that, if $\mathcal{P}$ is the set of pairs $\{s,t\}$ such that st is an edge of $\mathcal{T}^1$, the vertices built by the algorithm above are the centroids of the elements of $\mathcal{P}$. Moreover, since the edges of $\mathcal{T}^1$ are pairwise disjoint except at their endpoints, these centroids are pairwise disjoint.

We will study now the connected components of $g^2(V) \setminus \tau$. For this, let us consider an arbitrary point s of V. Let $t_1, \ldots, t_m$ be the neighbors of s in $\mathcal{T}^1$ in counter-clockwise direction around s, such that, if s is a vertex of $conv(V)$, t_1 and t_m are respectively the successor and the predecessor of s on $\delta(conv(V))$ (in the counter-clockwise direction). The midpoints of the line segments $st_1, st_2, \ldots, st_m$ are the vertices a the polygonal line $\mathcal{L}_s$ whose edges are edges of triangles of τ.

In the case where s is not a vertex of $conv(V)$, $\mathcal{L}_s$ is closed and is the boundary of a connected component of $g^2(V) \setminus \tau$.

When s is a vertex of $conv(V)$, $\mathcal{L}_s$ links the two points $g(\{s,t_1\})$ and $g(\{s,t_m\})$. Since st_1 and st_m are edges of $conv(V)$, $\{s,t_1\}$ and $\{s,t_m\}$ are respectively separable from $V \setminus \{s,t_1\}$ and from $V \setminus \{s,t_m\}$ by a straight line and, hence, $g(\{s,t_1\})$ and $g(\{s,t_m\})$ are vertices of $g^2(V)$. The part of $g^2(V)$ on the left of the polygonal line $\mathcal{L}_s$, oriented from $g(\{s,t_1\})$ to $g(\{s,t_m\})$, forms then, either an empty

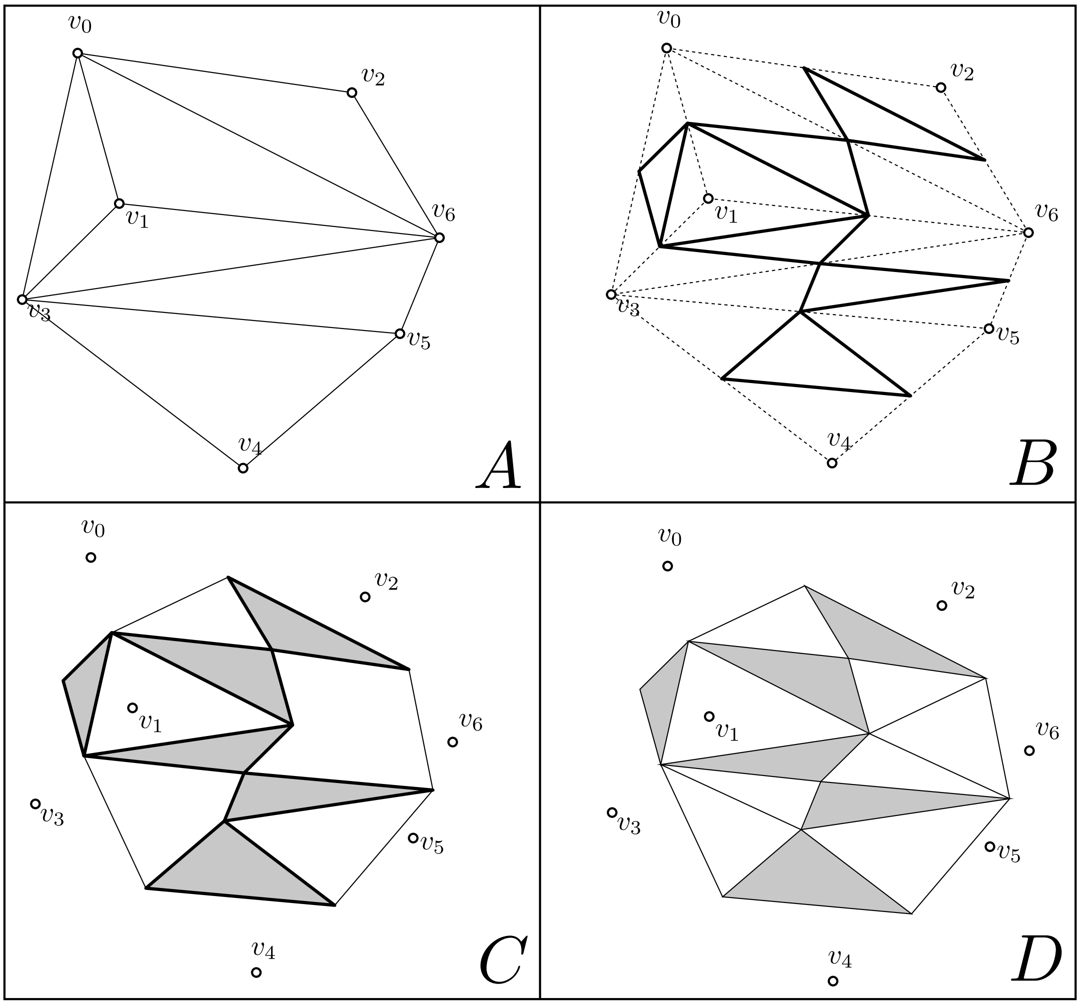

Figure 2.23: Illustration of the application of the `build_order_k_triangulation` algorithm on an (order-1) triangulation: For each triangle we build its corresponding 2-neighbor domain triangle (A and B). We later compute a constrained triangulation of the 2-set polygon without these domain triangles and we get an order-2 triangulation (C and D).

set, or one or more connected components of $g^2(V) \setminus \tau$. Moreover, since all the vertices of $g^2(V)$ are vertices of τ, all the vertices of these connected components are vertices of $\mathcal{L}_s$.

Conversely, every connected component of $g^2(V) \setminus \tau$ is delimited by a part of such a line $\mathcal{L}_s$, with s a point of V. It results that every triangle of a constrained triangulation of a connected component of $g^2(V) \setminus \tau$ is of the form $g(\{s, t_h\})g(\{s, t_i\})g(\{s, t_j\})$ and is then a territory triangle.

We can then give the following result:

Proposition 2.23. *When the algorithm* `build_order_k_triangulation` *is applied to an order-*1 *triangulation* $\mathcal{T}^1$ *of* V*, it constructs a* 2*-neighbor triangulation*

$\mathcal{T}^2$ *of* V*. Moreover, the triangles of* τ *are the domain triangles of* $\mathcal{T}^2$ *and* $g(\{s,t\})$ *is a vertex of* $\mathcal{T}^2$ *if, and only if,* st *is an edge of* $\mathcal{T}^1$*.*

Every triangulation built this way is called an order-2 (centroid) triangulation of V.

Remark 2.24. *Not every* 2*-neighbor triangulation is an order-*2 *triangulation, that is, it cannot be constructed from an order-*1 *triangulation using the algorithm* `build_order_k_triangulation` *(see Figure 2.24).*

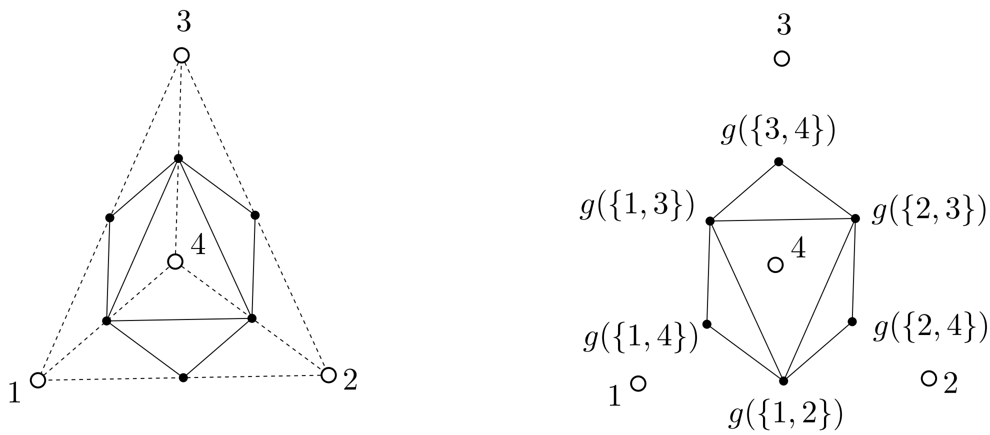

Figure 2.24: The set $V = \{1, 2, 3, 4\}$ admits a unique order-1 triangulation and a unique order-2 triangulation (left figure). The right figure shows a 2-neighbor triangulation of V that is not an order-2 triangulation of V.

Liu and Snoeyink proved that by applying the `build_order_k_triangulation` algorithm to an order-2 triangulation of V, we obtain in the same way a 3-neighbor triangulation of V whose domain triangles are triangles of τ (see Figure 2.25).

Practical experimentations done independently by Liu and Snoeying and by ourselves showed that by applying algorithm `build_order_k_triangulation` successively, we always obtain triangulations of the same type. However, this result is not proven for $k > 3$. So, we state the conjecture of Liu and Snoeying [LS07]:

Conjecture 2.25. *Starting with an arbitrary (order-*1*) triangulation of* V *and applying* $k-1$ *times the algorithm* `build_order_k_triangulation`*, we obtain a* k*-neighbor triangulation of* V *from which* τ *is the set of domain triangles.*

Every triangulation obtained this way is called an order-k centroid triangulation of V.

Note that, from Property 2.22, all the triangles of an order-$(n-1)$ triangulation are domain triangles. Since, in addition, $g^n(V)$ is reduced to the unique

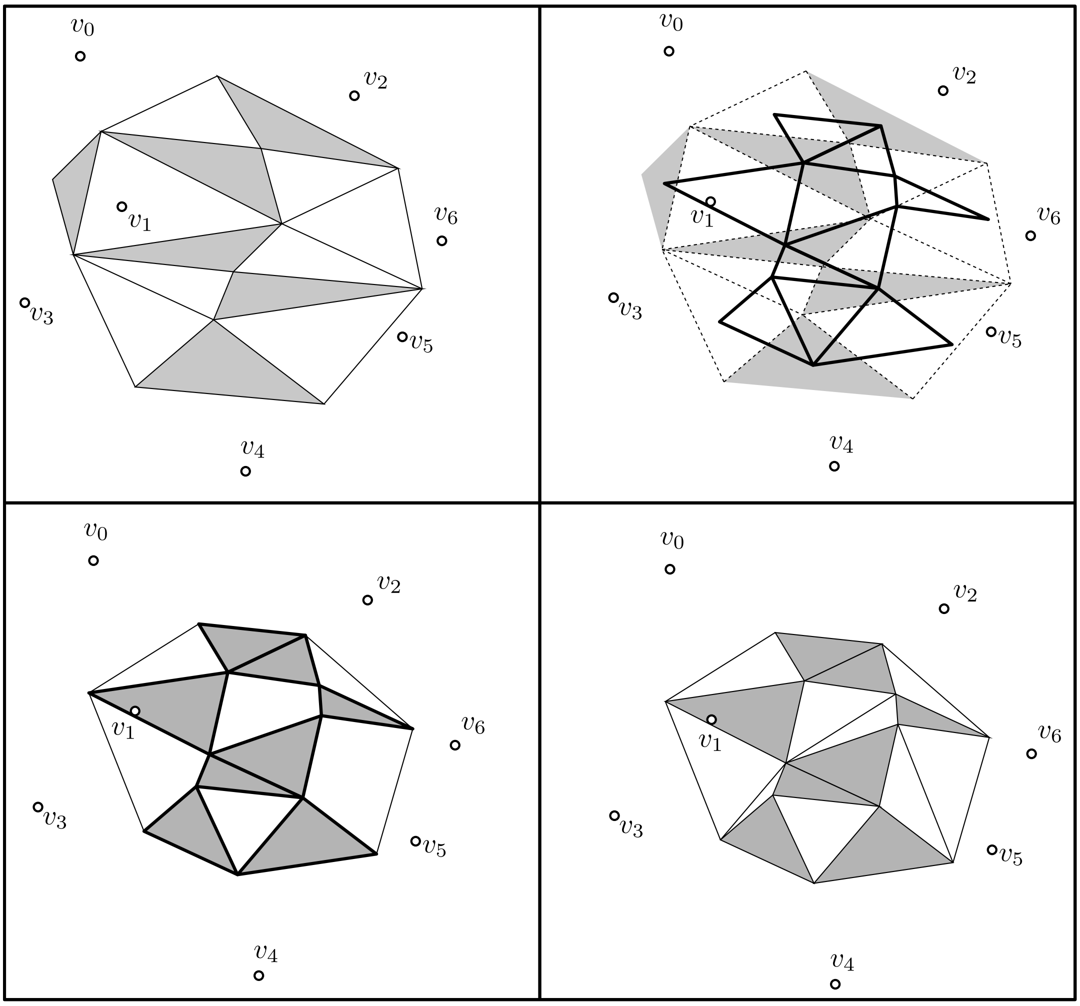

Figure 2.25: An order-3 triangulation obtained from an order-2 triangulation by algorithm `build_order_k_triangulation`.

point $g(V)$, the algorithm `build_order_k_triangulation` when applied to an order-$(n-1)$ triangulation does not generate any triangle. We will say then that V has a unique order-n triangulation which is reduced to $g^n(V) = g(V)$.

Liu and Snoeyink have also conjectured that the size of every order-k triangulation of a set V of n points is in $O(k(n-k))$, like the size of the order-k Delaunay triangulation.

Thus, since the construction of an (order-1) triangulation takes $\Omega(n \log n)$ time at least, we cannot hope building an order-k triangulation in less than $O(n \log n + k^2(n-k))$ time, if we apply $k-1$ times the `build_order_k_triangulation` algorithm.

Chapter 3

k-sets of convex inclusion chains

3.1 Introduction

In this chapter, a new k-set invariant is given. To this aim we introduce the convex inclusion chain of the set of points V as a sequence of the points of V such that no point is in the convex hull of the previous ones. The set of k-sets of a convex inclusion chain is the set of k-sets of all initial subsequences of this chain. We show that the number of these k-sets does not depend on the convex inclusion chain of V. Moreover, this number is equal to the number of regions of the order-k Voronoi diagram: $2kn - n - k^2 + 1 - \sum_{i=1}^{k-1} \gamma^i(V)$.

To get this result we use the k-set polygon introduced in Chapter 2. We show that, when a new point is added outside of the convex hull, the edges to remove from the current k-set polygon form a polygonal line and the edges to create to obtain the new k-set polygon form also a polygonal line.

We study first the particular case $k = 1$ where the k-set polygon is reduced to the convex hull. This case will serve us as a model for the general case.

3.2 Adding a point to a convex hull

We will start by giving a formal definition of a convex inclusion chain:

Definition 3.1. *A convex inclusion chain of the point set V is a sequence $(v_1, v_2,$ ..., $v_n)$ of the points of V, such that for every integer $i \in \{1, \ldots, n-1\}$, $v_{i+1} \notin$ conv$(\{v_1, \ldots, v_i\})$.*

From here on, we denote by V_i the set $\{v_1, \ldots, v_i\}$, for $i \in \{1, \ldots, n\}$.

If $k = 1$ the k-set polygon is equal to the convex hull.

Note that in the case of the convex hull of two points, the convex hull is made of two oriented edges, linking the vertices in both directions.

Assume that, for $i \geq 2$ the convex hull $conv(V_{i-1})$ of V_{i-1} is constructed. What we want to do now, is to add the new point v_i to $conv(V_{i-1})$ to get $conv(V_i)$. This update process, requires two essential steps:

First, we need to remove some edges from $conv(V_{i-1})$. Afterward, we will prove that what remains of $\delta(conv(V_{i-1}))$ after this step, forms a polygonal line.

The second step, involves adding some edges to what remained of $\delta(conv(V_{i-1}))$ to get $conv(V_i)$.

Eventually, once we know how to do these two steps, we can add the points one after the other to finally get $conv(V_n)$.

So the first thing to do is to characterize which are the edges to remove, and which ones to keep, once the point v_{i+1} is added.

Lemma 3.2. *An edge $v_r v_l$ of $conv(V_{i-1})$ is an edge of $conv(V_i)$ if and only if $v_i \in (v_r \overset{\circ}{} v_l)^+$.*

Proof. From Property 2.7, $v_r v_l$ is an edge of $conv(V_i)$ if and only if $(v_r v_l)$ has all the points of $V_i \setminus \{v_r, v_l\}$ on its left. Hence, since v_i is the only new point, an edge $v_r v_l$ of $conv(V_{i-1})$ is an edge of $conv(V_i)$ if and only if $v_i \in (v_r \overset{\circ}{} v_l)^+$. $\square$

Another important property is needed:

Lemma 3.3. *(i) Exactly two edges are going to be created to get $conv(V_i)$ from $conv(V_{i-1})$ and these two edges are consecutive on $\delta(conv(V_i))$.*

(ii) The edges of $\delta(conv(V_{i-1}))$ that are not edges of $\delta(conv(V_i))$ form a polygonal line.

Proof. (i) Obviously, since v_i belongs to V_i and not to $conv(V_{i-1})$, v_i is then a vertex of $conv(V_i)$. All other vertices of $conv(V_i)$ are vertices of $conv(V_{i-1})$ and every edge of $conv(V_i)$ is either an edge of $conv(V_{i-1})$, or links v_i to a vertex of $conv(V_{i-1})$. It results that exactly two consecutive edges are created when building $conv(V_i)$ and that all the other edges of $conv(V_i)$ form a connected polygonal line on $\delta(conv(V_{i-1}))$.

(ii) It follows that, the removed edges of $\delta(conv(V_{i-1}))$ form also a connected polygonal line. $\square$

Now that we have everything we need, the number of created edges can be obtained.

Theorem 3.4. *Any algorithm that incrementally constructs the convex hull of a convex inclusion chain $(v_1, \ldots, v_n)$ creates $2(n-1)$ edges.*

Proof. $conv(V_1)$ is a single point. From Lemma 3.3, for any added point v_i (for $i > 1$), two edges are constructed. Summing all constructed edges for the the remaining $n-1$ points will give us $2(n-1)$ created edges. $\square$

3.3 A new k-set invariant

In the previous section, we computed the number of edges we have to create if we add the points to a convex hull incrementally. What we have done before will serve us as a model to compute the total number of created k-sets if we add the points incrementally as well.

Similarly to what we have done previously, it will be assumed that the k-set polygon $g^k(V_{i-1})$ is constructed, for an $i \geq k$ and $k < n$. We update $g^k(V_{i-1})$ to get $g^k(V_i)$ and to this aim, the edges to remove from $g^k(V_{i-1})$ once the point v_i is added, are characterized. The newly created edges are characterized as well (see Figure 3.1).

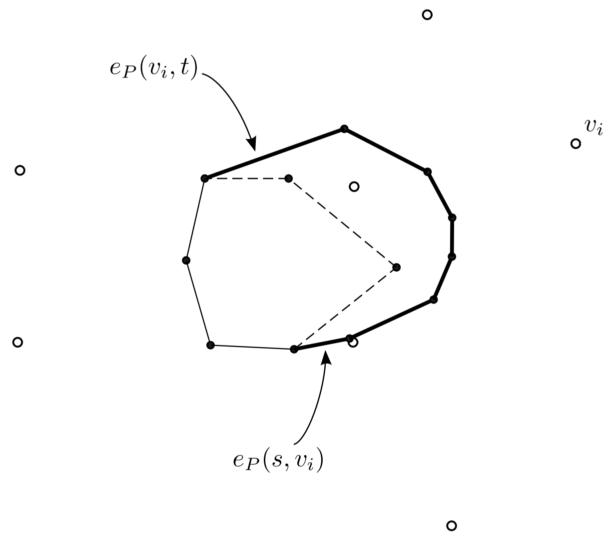

Figure 3.1: The updating of $g^k(V_{i-1})$ when v_i is added: The edges to be removed are in dashed lines and the edges to create are in bold lines.

Lemma 3.5. *An edge $e_P(s,t)$ of $g^k(V_{i-1})$ is an edge of $g^k(V_i)$ if and only if $v_i \in (\overset{\circ}{s}t)^+$.*

Proof. From Proposition 2.12, any edge $e_P(s,t)$ of the k-set polygon must have $k-1$ points (the subset P) on the right of (st). If v_i is also on the right of (st), then the edge $e_P(s,t)$ is not an edge of $g^k(V_i)$.

In the opposite case, if v_i is on the left of (st) then, in this case $e_P(s,t)$ is an edge of $g^k(V_i)$ because (st) still has $k-1$ points on its right (from Proposition 2.12). $\square$

Lemma 3.6. *(i) An edge $e_P(s,t)$ of $g^k(V_i)$ is not an edge of $g^k(V_{i-1})$ if and only if $v_i \in P \cup \{s,t\}$.*

(ii) $g^k(V_i)$ admits one and only one edge $e_P(s,t)$ that has $s = v_i$ (resp. $t = v_i$).

Proof. (i) Obviously, if $v_i \in P \cup \{s,t\}$ then $P \cup \{s,t\} \not\subseteq V_{i-1}$ and $e_P(s,t)$ is not an edge of $g^k(V_{i-1})$. On the other hand, if $v_i \notin P \cup \{s,t\}$ then $P \cup \{s,t\} \subseteq V_{i-1}$, $V_{i-1} \cap (\overset{\circ}{st})^- = P$ and $e_P(s,t)$ is an edge of $g^k(V_{i-1})$.

(ii) Since $v_i \notin conv(V_{i-1})$ and no three points of V_i are collinear, there is one and only one point t of V_{i-1} such that $|(\overset{\circ}{v_i t})^+ \cap V_{i-1}| = k-1$ (see Figure 3.2). Thus, from Proposition 2.12, there is one and only one edge $e_P(s,t)$ of $g^k(V_i)$ such that $s = v_i$. In the same way there is one and only one edge of the form $e_P(s, v_i)$.

□

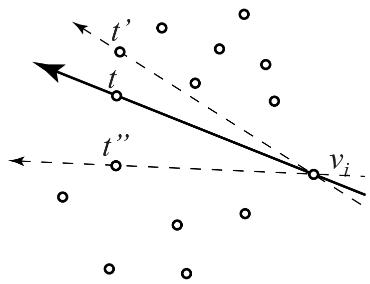

Figure 3.2: Illustration of the proof for Lemma 3.6

The new added point v_i can be separated from V_{i-1} by a straight line. There exists another important property that relates this straight line to the k-set polygon $g^k(V_{i-1})$.

Lemma 3.7. *(i) For any straight line Δ that strictly separates v_i from V_{i-1} and that is not parallel to any straight line passing through any two points of V_{i-1}, there is a unique vertex $g(T_{min})$ (resp. $g(T_{max})$) of $g^k(V_{i-1})$ closest to (resp. farthest from) Δ.*

(ii) If $|V_{i-1}| > k$, at least one of the edges of $g^k(V_{i-1})$ incident to $g(T_{min})$ (resp. $g(T_{max})$) is not (resp. is) an edge of $g^k(V_i)$.

Proof. (i) Let Δ be oriented such that $v_i \in \Delta^-$. Let Δ_1 be a straight line parallel to Δ, oriented in the same direction as Δ, and such that $\Delta_1 \cap V_{i-1} = \emptyset$ and $|\Delta_1^- \cap V_{i-1}| = k$ (see Figure 3.3). Let $T_{min} = \Delta_1^- \cap V_{i-1}$ and let Δ_2 be the straight line parallel to Δ_1, passing through $g(T_{min})$, and oriented in the same direction as Δ_1. For every subset $T \neq T_{min}$ of k points of V_{i-1}, at least one point of T belongs to $\overset{\circ}{\Delta}{}_1^+$ and $g(T)$ belongs to $\overset{\circ}{\Delta}{}_2^+$ from Lemma 2.9. Thus $g(T_{min})$ is the point of $g^k(V_{i-1})$ closest to Δ and is unique.

Let Δ_1' be an oriented straight line parallel to Δ such that ${\Delta'}_1 \cap V_{i-1} = \emptyset$ and $|{\Delta'}_1^+ \cap V_{i-1}| = k$. Let $T_{max} = {\Delta'}_1^+ \cap V_{i-1}$. In the same way, we can prove that the point $g(T_{max})$ of $g^k(V_{i-1})$ is farthest from Δ and is unique.

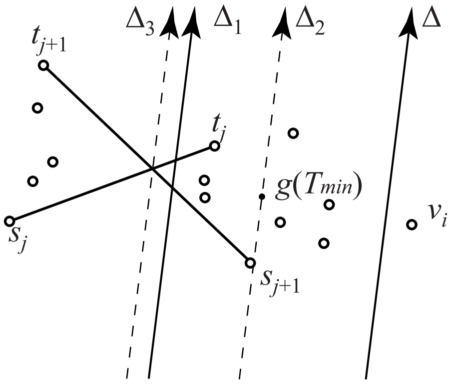

Figure 3.3: Illustration of Lemma's 3.7 proof with $k = 8$

(ii) Let $e_{P_j}(s_j, t_j)$ and $e_{P_{j+1}}(s_{j+1}, t_{j+1})$ be the two edges of $g^k(V_{i-1})$ with endpoint $g(T_{min})$ such that $P_j \cup \{t_j\} = P_{j+1} \cup \{s_{j+1}\} = T_{min}$. Since $\{s_{j+1}, t_j\} \subseteq T_{min} \subset \Delta_1^-$, since $\{s_j, t_{j+1}\} \subseteq S \setminus T_{min} \subset \Delta_1^+$, and since $s_jt_j \cap s_{j+1}t_{j+1} \neq \emptyset$ from Corollary 2.13, the straight line Δ_3 parallel to Δ_1, oriented in the same direction as Δ_1, and that passes through $s_jt_j \cap s_{j+1}t_{j+1}$, is such that $\Delta_3^- \subset (s_jt_j)^- \cup (s_{j+1}t_{j+1})^-$ (see Figure 3.3). Since $v_i \in \Delta^- \subset \mathring{\Delta}_3^-$, it follows that v_i belongs to at least one of the half planes $(\mathring{s_jt_j})^-$ or $(\mathring{s_{j+1}t_{j+1}})^-$ and thus, from Proposition 2.12, at least one of the edges $e_{P_j}(s_j, t_j)$ and $e_{P_{j+1}}(s_{j+1}, t_{j+1})$ is not an edge of $g^k(V_i)$.

In the same way, let $e_{P'_j}(s'_j, t'_j)$ and $e_{P'_{j+1}}(s'_{j+1}, t'_{j+1})$ be the two edges of $g^k(V_{i-1})$ with the endpoint $g(T_{max})$ such that $P'_j \cup \{t'_j\} = P'_{j+1} \cup \{s'_{j+1}\} = T_{max}$. Let Δ'_3 be the straight line parallel to Δ and with the same direction, passing through $s'_jt'_j \cap s'_{j+1}t'_{j+1}$. Since $\Delta_3'^- \subset (s'_jt'_j)^+ \cup (s'_{j+1}t'_{j+1})^+$ and since $v_i \in \mathring{\Delta}_3^- \subset \mathring{\Delta}_3'^-$, then v_i belongs to at least one of the half planes $(\mathring{s'_jt'_j})^+$ or $(\mathring{s'_{j+1}t'_{j+1}})^+$. Thus at least one of the edges $e_{P'_j}(s'_j, t'_j)$ and $e_{P'_{j+1}}(s'_{j+1}, t'_{j+1})$ is an edge of $g^k(V_i)$. $\square$

Now, we have proved the existence of at least one removed edge of $g^k(V_{i-1})$, and the existence of at least one new created edge to get $g^k(V_i)$. This makes it possible for us, to characterize the newly created edges and the removed ones on $g^k(V_{i-1})$ to get $g^k(V_i)$.

For the next proposition, we will assume that $|V_{i-1}| > k$.

Proposition 3.8. *(i) The edges of $g^k(V_i)$ that are not edges of $g^k(V_{i-1})$ form an open connected polygonal line with at least two edges, whose first (resp. last) edge in counter clockwise direction is the unique edge of $g^k(V_i)$ of the form $e_P(s,t)$ with $t = v_i$ (resp. $s = v_i$).*

(ii) The edges of $g^k(V_{i-1})$ that are not edges of $g^k(V_i)$ form an open connected and non empty polygonal line.

Proof. From Lemma 3.6, the set $\mathcal{C}$ of edges of $\delta(g^k(V_i))$ that are not edges of $\delta(g^k(V_{i-1}))$ admits at least two edges. Also from Lemma 3.7 there exists at least one edge of $\delta(g^k(V_{i-1}))$ that is an edge of $\delta(g^k(V_i))$ too. Thus, $\mathcal{C}$ admits at least one edge $e_P(s,t)$ whose first endpoint is a vertex of $\delta(g^k(V_{i-1}))$ (*i.e.* $v_i \notin P \cup \{s\}$). Hence, from Lemma 3.6, $t = v_i$ and $e_P(s,t)$ is the only edge of $\delta(g^k(V_i))$ of the form $e_P(s, v_i)$. In the same way, there is a unique edge of $\mathcal{C}$ whose second endpoint is a vertex of $\delta(g^k(V_{i-1}))$ and this edge is of the form $e_P(v_i, t)$.

Since there is a unique edge of $\delta(g^k(V_i))$ that links, in the counter clockwise direction, a vertex of $\delta(g^k(V_i)) \setminus \delta(g^k(V_{i-1}))$ to a vertex of $\delta(g^k(V_i)) \cap \delta(g^k(V_{i-1}))$ and a unique edge of $\delta(g^k(V_i))$ that links a vertex of $\delta(g^k(V_i)) \cap \delta(g^k(V_{i-1}))$ to a vertex of $\delta(g^k(V_i)) \setminus \delta(g^k(V_{i-1}))$, it follows that every other edge of $g^k(V_i)$, either links two vertices of $\delta(g^k(V_i)) \cap \delta(g^k(V_{i-1}))$ between them, or links two vertices of $\delta(g^k(V_i)) \setminus \delta(g^k(V_{i-1}))$ between them.

As a result, the edges of $\delta(g^k(V_i)) \setminus \delta(g^k(V_{i-1}))$ form a connected polygonal line. Moreover, from Lemma 3.7 this polygonal line is connected. It follows that the edges of $\delta(g^k(V_{i-1})) \setminus \delta(g^k(V_i))$ form a connected polygonal line.

□

This proposition generalizes the Lemma 3.3 which corresponds to the case $k = 1$.

For every $k \in \{1, ..., n-1\}$ and for every $i \in \{k+1, ..., n\}$, let c_i^k denote the number of edges of $g^k(V_i)$ that are not edges of $g^k(V_{i-1})$, i.e. the number of edges to create while constructing the k-set polygon of $V_i = V_{i-1} \cup \{v_i\}$ from the k-set polygon of V_{i-1}. Since the number of edges of $g^k(V_k)$ is zero, $c^k = \sum_{i=k+1}^{n} c_i^k$ is the total number of edges to be created by an algorithm that incrementally constructs $g^k(V)$ by successively determining $g^k(V_k)$, $g^k(V_{k+1})$, ..., $g^k(V_n)$.

For every $j \in \{1, ..., n-1\}$, the number of edges of the j-set-polygon of V is equal to the number of its vertices and thus to the number $\gamma^j(V)$ of k-sets of V, from Proposition 2.11.

Proposition 3.9. $c^1 = 2(n-1)$ *and* $c^k = k(2n-k-1) - \sum_{j=1}^{k-1} \gamma^j(V)$ *if* $1 < k < n$.

Proof. From Lemma 3.6, for every $i \in \{k+1, ..., n\}$, $g^k(V_i)$ admits at least two edges that are not edges of $g^k(V_{i-1})$. These two edges are of the form $e_Q(v_i, t)$ and $e_P(s, v_i)$. All other edges of $g^k(V_i)$ that are not edges of $g^k(V_{i-1})$ are of the form $e_{P'}(s', t')$ with $v_i \in P'$. If $k = 1$, no such other edge exists since $P = \emptyset$. Thus $c_i^1 = 2$, for every $i \in \{2, ..., n\}$, and

$$c^1 = \sum_{i=2}^{n} c_i^1 = 2(n-1) \ .$$

If $k \in \{2, ..., n-1\}$, from Lemma 3.5, $e_{P'}(s', t')$ is an edge of $g^k(V_i)$ with $v_i \in P'$ if and only if $e_{P' \setminus \{v_i\}}(s', t')$ is an edge of $g^{k-1}(V_{i-1})$ and is not an edge of $g^{k-1}(V_i)$. Thus, denoting by d_i^{k-1} the number of edges of $g^{k-1}(V_{i-1})$ that are not edges of $g^{k-1}(V_i)$, we have $c_i^k = 2 + d_i^{k-1}$. It follows that

$$c^k = \sum_{i=k+1}^{n} c_i^k = 2(n-k) + \sum_{i=k+1}^{n} d_i^{k-1} \ .$$

Now, since the number of edges of $g^{k-1}(V_{k-1})$ is zero, we have $d_k^{k-1} = 0$ and $\sum_{i=k+1}^{n} d_i^{k-1}$ is the total number of edges to be deleted by an algorithm that incrementally constructs $g^{k-1}(V)$ by successively determining $g^{k-1}(V_{k-1})$, $g^{k-1}(V_k)$, ... , $g^{k-1}(V_n)$. And since $\gamma^{k-1}(V)$ is equal to the total number of created edges minus the number of the removed ones:

$$\sum_{i=k+1}^{n} d_i^{k-1} = c^{k-1} - \gamma^{k-1}(V)$$

and

$$c^k = 2(n-k) + c^{k-1} - \gamma^{k-1}(V) \ .$$

Solving this induction relation, the following is obtained

$$c^k = (k-1)(2n-k-2) + c^1 - \sum_{j=1}^{k-1} \gamma^j(V) = k(2n-k-1) - \sum_{j=1}^{k-1} \gamma^j(V) \ .$$

□

This proposition shows that the number of edges that have to be created for the incremental construction of a k-set polygon -provided that every new inserted point does not belong to the convex hull of the previously inserted ones- does not depend on the order in which the points are treated. In addition, since $\sum_{j=1}^{k-1} \gamma^j(V)$ is the number of $(\leq (k-1))$-sets of V and since this number is known to be bounded by $(k-1)n$ (see [Pec85]), it follows that:

Corollary 3.10. *Any algorithm that incrementally constructs the k-set polygon of n points, so that no point belongs to the convex hull of the points inserted before him, has to create $\Omega(k(n-k))$ edges.*

The set of k-sets of $V_k, V_{k+1}, \ldots, V_n$ is called the set of k-sets of the convex inclusion chain $(v_1, \ldots, v_n)$ of V. Note that V_k is the unique k-set of V_k and since $v_{k+1} \notin conv(V_k)$ when $k < n$, V_k is also a k-set of V_{k+1}.

Lemma 3.11. *T is a k-set of the convex inclusion chain $(v_1, \ldots, v_n)$ if and only if there exists an integer $i \in \{k, \ldots, n\}$ such that $g(T)$ is a vertex of $g^k(V_i)$. Moreover, if T' is a k-set of $(v_1, \ldots, v_n)$ distinct from T then $g(T) \neq g(T')$.*

Proof. By definition, T is a k-set of $(v_1, \ldots, v_n)$ if and only if there exists an integer $i \in \{k, \ldots, n\}$ such that T is a k-set of V_i, that is, from Proposition 2.11, if and only if $g(T)$ is a vertex of $g^k(V_i)$.

If T' is a k-set of $(v_1, \ldots, v_n)$ distinct from T, there exists an integer $j \in \{k, \ldots, n\}$ such that T' is a k-set of V_j, that is $g(T')$ is a vertex of $g^k(V_j)$. We can suppose, without loss of generality, that $j \geq i$ and in this case $T \subseteq V_j$. If T is a k-set of V_j, $g(T)$ is a vertex of $g^k(V_j)$ distinct from $g(T')$, from Proposition 2.11. Otherwise, from the same proposition, $g(T)$ is not a vertex of $g^k(V_j)$ and is also distinct from $g(T')$. □

Using the previous results, it is easy to find the number of k-sets of a convex inclusion chain of V:

Theorem 3.12. *Any convex inclusion chain of a planar set V of n points admits $2kn - n - k^2 + 1 - \sum_{j=1}^{k-1} \gamma^j(V)$ k-sets (with $\sum_1^0 = 0$).*

Proof. From Lemma 3.11, the number of k-sets of a convex inclusion chain $(v_1, \ldots, v_n)$ of V is equal to the number of distinct k-set polygon vertices created by an incremental algorithm that successively constructs $g^k(V_{k+1})$, ... , $g^k(V_n)$. The number of vertices of $g^k(V_{k+1})$ is equal to the number c_{k+1}^k of its edges. Moreover, from proposition 3.8, for every $i \in \{k+2, ..., n\}$, the edges of $g^k(V_i)$ that are not edges of $g^k(V_{i-1})$ form an open connected and non empty polygonal line. Thus, the number of vertices of this line that are not vertices of $g^k(V_{i-1})$ is $c_i^k - 1$, where c_i^k is the number of edges of the line. It follows that the number of k-sets of V is $c_{k+1}^k + \sum_{i=k+2}^n (c_i^k - 1)$, that is, from Proposition 3.9, $2kn - n - k^2 + 1 - \sum_{j=1}^{k-1} \gamma^j(V)$ since $c^k = \sum_{i=k+1}^n c_i^k$. □

According to this theorem, the number of k-sets of a convex inclusion chain of a set V only depends on the set V and not on the chosen chain. An even more intriguing consequence of the theorem arises from its connection with order-k Voronoi diagrams. Actually, as we have seen in Chapter 2, the number of regions in the order-k Voronoi diagram of V is also equal to $2kn - n - k^2 + 1 - \sum_{j=1}^{k-1} \gamma^j(V)$. Since a subset of k points of V generates an order-k Voronoi region if and only if it can be separated from the remaining points by a circle, it follows that:

Corollary 3.13. *Given a set V of points in the plane, no three of them being collinear and no four of them being co-circular, the number of k-sets of a convex inclusion chain of V is equal to the number of subsets of k points of V that can be separated from the remaining by a circle.*

This result is very surprising since the set of k-sets of a convex inclusion chain depends on the chosen chain. One might wonder if the subsets of k points of V

separable from the rest by a circle are the k-sets of a particular convex inclusion chain of V. In other words, does every set V have a convex inclusion chain whose every k-set is separable from the rest of the points of V by a circle?

The following example proves that it is not always the case (see Figure. 3.4).

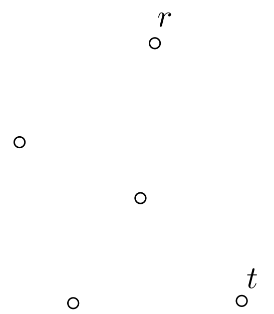

Figure 3.4: This point set admits no convex inclusion chain whose every 2-set can be separated from the other points by a circle

Let V be a set of six points, five of them being the vertices of a regular pentagon $\mathcal{P}$ and the sixth being placed in the center of the circle circumscribed to $\mathcal{P}$. We can slightly move the vertices of $\mathcal{P}$ so that we do not have more than three co-circular vertices.

By definition, the last element of every convex inclusion chain $\mathcal{V}$ of V is a vertex of $conv(V)$, that is, a vertex s of $\mathcal{P}$. The two neighbors r and t of s on $\mathcal{P}$ form then an edge of $conv(V \setminus \{s\})$ and, therefore, a 2-set of $V \setminus \{s\}$. The set $\{r, t\}$ is then a 2-set of $\mathcal{V}$ but it is not separable from V by a circle. It results that V has no convex inclusion chain whose every 2-set is separable from the rest of points of V by a circle.

In chapter 6, we will try to explain the result of Corollary 3.13, by establishing other links between k-sets of convex inclusion chains and order-k Voronoi diagrams.

Chapter 4

Incremental construction algorithm

4.1 Introduction

The incremental construction of the convex hull of a set of points in the plane is a simple algorithm which seems to have existed long time before it was mentioned by Preparata and Shamos [PS85] and by Kallay [Kal84] where it was generalized to higher dimensions.

In [Mel87], Melkman showed that the convex hull of a simple polygonal line can be constructed on-line in linear time. In this chapter we give an on-line algorithm that incrementally constructs the k-set polygon of a special convex inclusion chain that is also a simple polygonal line.

This chapter starts by illustrating the idea of the algorithm for the special case $k = 1$ where the k-set polygon is the convex hull. This algorithm is a particular case of Melkman's algorithm.

Later on, the algorithm is generalized for the case $k \geq 1$. The results obtained in the previous chapter are going to be used to compute the complexity of the algorithm.

4.2 Convex hull of a special convex inclusion chain

We first give an incremental convex hull construction algorithm for a special convex inclusion chain $(v_1, \ldots, v_n)$ of V that is also a simple polygonal line.

There are two steps involving the update of the convex hull of the set $V_{i-1} = \{v_1, \ldots, v_{i-1}\}$ to get the convex hull of the set $V_i = V_{i-1} \cup \{v_i\}$. The first step is about finding the two extrema vertices on $\delta(conv(V_{i-1}))$, the first extremum in the clockwise direction and the other extremum in the counter clockwise direction. In the second step, we connect the point v_i to both extrema via two edges.

Since the edges to remove from $\delta(conv(V_{i-1}))$ form a connected polygonal line, it suffices to find one vertex of the edges to remove. Once such a vertex is found we iterate through the edges that have the point v_i on their right and remove them.

To find a first vertex, we prove that the vertex v_{i-1} is actually a vertex on the removed polygonal line.

Lemma 4.1. *If $i \geq 3$, the vertex v_{i-1} of $conv(V_{i-1})$ and at least one edge of $conv(V_{i-1})$ incident in v_{i-1} are visible from the point v_i.*

Proof. Suppose that the line segment $v_{i-1}v_i$ cuts an edge $v_s v_t$ of $conv(V_{i-1})$. Then, the point v_{i-1}, which is also a vertex of $conv(V_{i-1})$, is not visible from v_i and, therefore, it is also a vertex of $conv(V_i)$. Since v_i is also a vertex of $conv(V_i)$ and since the part of the polygonal line $(v_1, \ldots, v_i)$ between the two vertices v_s and v_t lies inside $conv(V_{i-1})$ then, $v_{i-1}v_i$ cuts necessarily an edge of the polygonal line but this is impossible. This means that v_{i-1} is visible from v_i and it is the same for at least one edge of $conv(V_{i-1})$ incident in v_{i-1}. □

This means that we can find a first vertex on the polygonal line to remove in constant time. This also implies that we can find the whole polygonal line to remove in a time proportional to the number of edges to remove on $\delta(conv(V_{i-1}))$. Thus the algorithm:

```
function construct_convex_hull(v_1,...,v_n)
{
  construct the convex hull of V_2;

  for( i=3 to n )
  {
    Remove all the edges of conv(V_{i-1}) visible from v_i starting
        by the edges incident in the point v_{i-1};

    Link v_i to the endpoints of the remaining polygonal line;
  }
  return the convex hull;
}
```

Theorem 4.2. *The algorithm that constructs incrementally the convex hull of a convex inclusion chain that is also a simple polygonal line, performs in $O(n)$ time.*

Proof. Finding the polygonal line to remove once v_i is added, requires iterating through the edges of $conv(V_{i-1})$ and testing them against v_i, starting from the edges incident in v_{i-1}. This is done in two steps: In the first step we start looking for the edges that have the point v_i on their right, in the counter clockwise direction, and in the second one, we restart the same process from v_{i-1} in the clockwise direction this time. These two steps require a time proportional to the the total number of edges to remove. However, we cannot remove more edges than we can create. Since we create exactly two edges at each step, the total running time of the algorithm is $O(n)$. $\square$

Remark 4.3. *If we want to generalize the algorithm in the case where V_n is a convex inclusion chain without being a simple polygonal line, we cannot find a first vertex visible from the point v_i in constant time.*

4.3 Incremental construction of k-sets

In this section we further study the edges to remove and create to update the k-set polygon when a new point is added outside the convex hull of the current set of points.

To this aim, we need to generalize the notion of separability.

Given an oriented straight line Δ and a set V, we say that a set T is Δ-separable from V if T is a subset of V such that $\mathring{\Delta}^- \cap V = T$. Moreover, T is said to be $//_\Delta$-separable from V if there exists a straight line Δ', parallel to Δ and oriented as Δ, such that T is Δ'-separable from V.

For the sake of simplicity we say that a vertex of a convex polygon $\mathcal{P}$ is Δ-separable (resp. $//_\Delta$-separable) from $\mathcal{P}$ if it is Δ-separable (resp. $//_\Delta$-separable) from the vertices of $\mathcal{P}$.

Let $(v_1, \ldots, v_n)$ be a convex inclusion chain of V and, for all $i \in \{1, \ldots, n\}$ let $V_i = \{v_1, \ldots, v_i\}$. For all $i \in \{k+2, \ldots, n\}$, we will note by $\mathcal{D}_i$ the set of edges of $g^k(V_{i-1})$ that are not edges of $g^k(V_i)$, and by $\mathcal{C}_i$ the set of edges of $g^k(V_i)$ that are not edges of $g^k(V_{i-1})$. $\mathcal{D}_i$ and $\mathcal{C}_i$ are actually polygonal lines from Proposition 3.8. In the particular case $i = k+1$, $\mathcal{D}_i$ is reduced to the centroid $g^k(V_k)$ and $\mathcal{C}_i$ is the whole boundary of $g^k(V_{k+1})$.

Also we note by $T_1, \ldots, T_m$ the k-sets of V_{i-1} such that $(g(T_1), \ldots, g(T_m))$ is the sequence of vertices of $\mathcal{D}_i$ in the counter clockwise direction (see Figure 4.1).

For every vertex $g(T_j)$ of $\mathcal{D}_i$, let $\mathcal{C}_{i,j}$ be the set:

- of vertices $g(T)$ of $\mathcal{C}_i$ such that $g(T)$ and $g(T_j)$ are respectively $//_\Delta$-separable from $g^k(V_i)$ and from $g^k(V_{i-1})$, for some straight line Δ (see Figure 4.2 and 4.3)

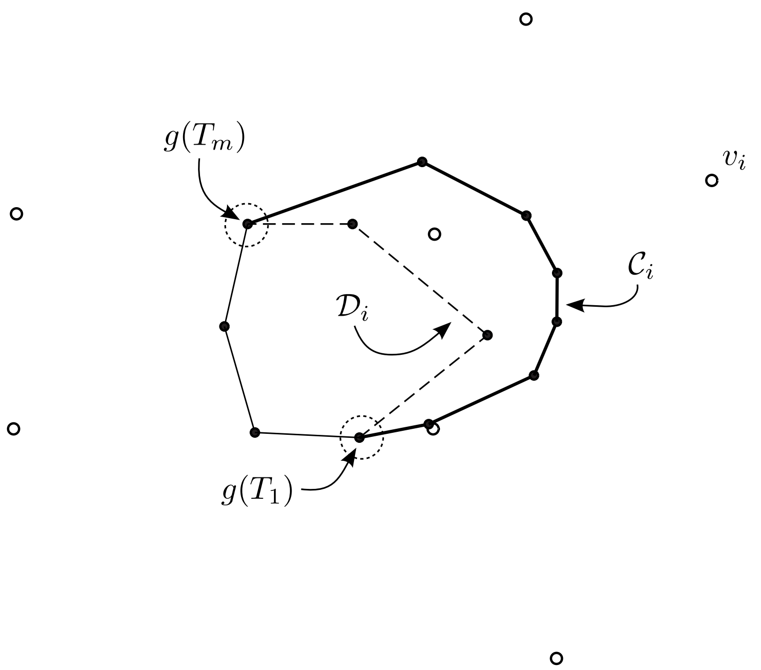

Figure 4.1: The polygonal line $\mathcal{D}_i$ to remove from $g^4(V_{i-1})$ and the polygonal line to create $\mathcal{C}_i$ when building $g^4(V_i)$, once the point v_i is added to $g^4(V_{i-1})$.

- and of edges of $\mathcal{C}_i$ that connect these vertices (see Figure 4.2).

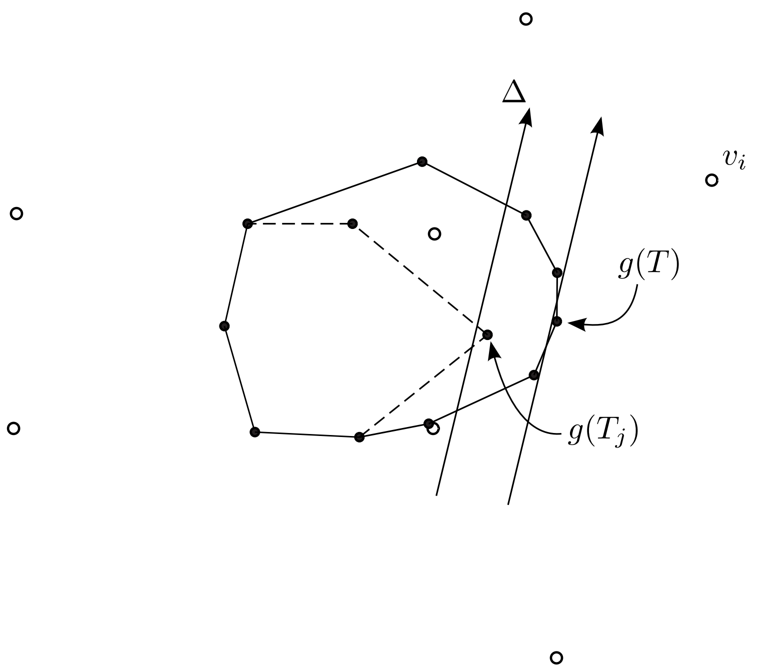

Figure 4.2: $g(T_j)$ and $g(T)$ are $//_\Delta$-separable from $g^4(V_{i-1})$ and $g^4(V_i)$ respectively.

Lemma 4.4. *For every $j \in \{1, \ldots, m\}$, $\mathcal{C}_{i,j}$ is a connected polygonal line.*

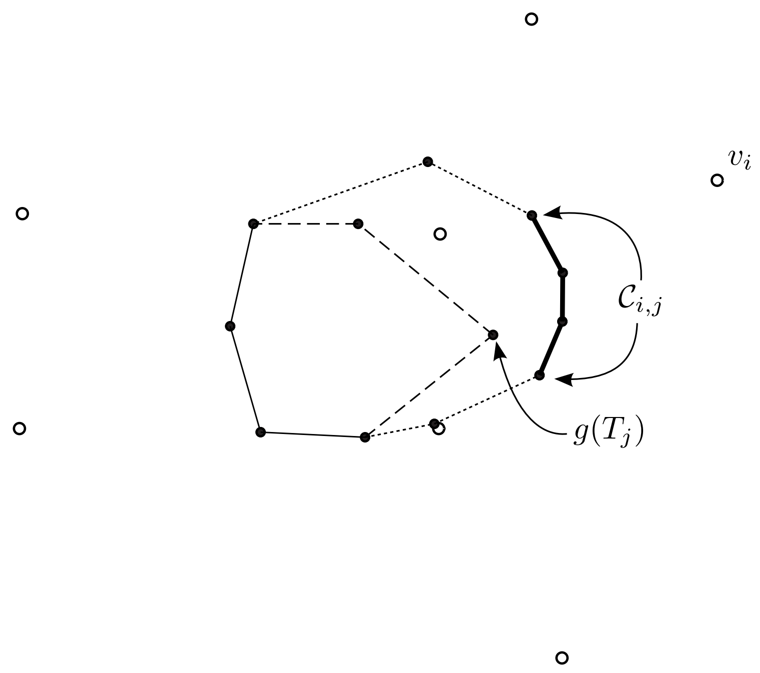

Figure 4.3: The part $\mathcal{C}_{i,j}$ of $\mathcal{C}_i$ associated with the vertex $g(T_j)$ of $\mathcal{D}_i$.

Proof. (i) If $g^k(V_{i-1})$ is not reduced to the single vertex $g(T_j)$, let Δ_j and Δ_{j+1} be the two oriented straight lines that support the two edges respectively entering and leaving $g(T_j)$. The oriented straight lines Δ for which $g(T_j)$ is Δ-separable from $g^k(V_{i-1})$ are the ones that cut the open edges of $g^k(V_{i-1})$ incident in $g(T_j)$ and such that $g(T_j) \in \Delta^-$. Hence, $g(T_j)$ is $//_\Delta$-separable from $g^k(V_{i-1})$ if and only if $0 < \measuredangle(\Delta_j, \Delta) < \measuredangle(\Delta_j, \Delta_{j+1})$. Since the vertices of $\mathcal{C}_{i,j}$ are the vertices of the convex polygonal line $\mathcal{C}_i$ that are $//_\Delta$-separable from $g^k(V_i)$, for these same oriented straight lines Δ, it results that $\mathcal{C}_{i,j}$ is a connected part of $\mathcal{C}_i$.

(ii) If $g^k(V_{i-1})$ is reduced to the single vertex $g(T_j)$, this vertex is $//_\Delta$-separable from $g^k(V_{i-1})$ no matter how Δ is positionned in the plane. It results that $\mathcal{C}_{i,j} = \mathcal{C}_i = \delta(g^k(V_i))$ is connected. □

Lemma 4.5. *If $\mathcal{C}_{i,j}$ is oriented in the counter clockwise direction on $\mathcal{C}_i$ then:*

(i) $g(T_1)$ is the first vertex of $\mathcal{C}_{i,1}$,

(ii) $g(T_m)$ is the last vertex of $\mathcal{C}_{i,m}$,

(iii) for every $j \in \{2, \ldots, m\}$, if $e_{P_j}(s_j, t_j)$ is the edge of $\mathcal{D}_i$ that links $g(T_{j-1})$ and $g(T_j)$, then $g(P_j \cup \{v_i\})$ is the last vertex of $\mathcal{C}_{i,j-1}$ and the first vertex of $\mathcal{C}_{i,j}$.

Proof. (i) Since $g(T_1)$ is a vertex of $g^k(V_i)$, there exists an oriented straight line Δ such that $g(T_1)$ is Δ-separable from $g^k(V_i)$. Since $T_1 \subseteq V_{i-1} \subset V_i$, $g(T_1)$ is also Δ-separable from $g^k(V_{i-1})$. It results that $g(T_1)$ is a vertex of $\mathcal{C}_{i,1}$ and, since $g(T_1)$ is the first vertex of $\mathcal{C}_i$, it is also the first vertex of $\mathcal{C}_{i,1}$.

(ii) In the same way, there exists an oriented straight line Δ such that $g(T_m)$ is Δ-separable from $g^k(V_i)$ and from $g^k(V_{i-1})$. $g(T_m)$ is then the last vertex of $\mathcal{C}_{i,m}$.

(iii) Since $e_{P_j}(s_j, t_j)$ is an edge of $\mathcal{D}_i$, $v_i \in (\overset{\circ}{s_j t_j})^-$, from Lemma 3.5. It results that $P_j \cup \{v_i\}$ is $(s_j t_j)$-separable from V_i and thus $g(P_j \cup \{v_i\})$ is a vertex of $\mathcal{C}_i$. Since V_i is finite, $P_j \cup \{v_i\}$ is also $//_\Delta$-separable from V_i for every oriented straight line Δ such that the angle $\measuredangle((s_j t_j), \Delta)$ tends toward 0. From the Proposition 2.11, $g(P_j \cup \{v_i\})$ is also $//_\Delta$-separable from $g^k(V_i)$. Now, for such oriented straight lines Δ with $\measuredangle((s_j t_j), \Delta) > 0$, $g(P_j \cup \{t_j\}) = g(T_j)$ is $//_\Delta$-separable from $g^k(V_{i-1})$, since $(s_j t_j)$ is parallel to $e_{P_j}(s_j, t_j)$. It results that $g(P_j \cup \{v_i\})$ is a vertex of $\mathcal{C}_{i,j}$. Moreover, since all oriented straight lines Δ' such that $g(T_j)$ is $//_{\Delta'}$-separable from $g^k(V_{i-1})$ are such that $\measuredangle((s_j t_j), \Delta') > 0$, $g(P_j \cup \{v_i\})$ is the first vertex of $\mathcal{C}_{i,j}$.

In the same way, $g(P_j \cup \{v_i\})$ and $g(T_{j-1}) = g(P_j \cup \{s_j\})$ are $//_\Delta$-separable from $g^k(V_i)$ and $g^k(V_{i-1})$ respectively when $\measuredangle((s_j t_j), \Delta) < 0$. It results that $g(P_j \cup \{v_i\})$ is also the last vertex of $\mathcal{C}_{i,j-1}$ (see Figure 4.4). $\square$

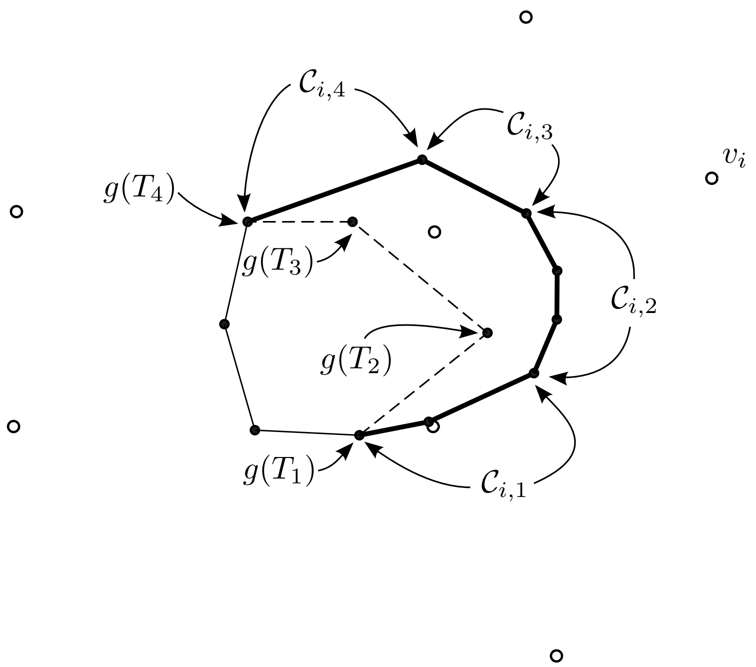

Figure 4.4: $\mathcal{C}_i$ is the sequence of polygonal lines $\mathcal{C}_{i,1}$, $\mathcal{C}_{i,2}$, $\mathcal{C}_{i,3}$, $\mathcal{C}_{i,4}$

Corollary 4.6. *$\mathcal{C}_i$ is the sequence of polygonal lines $\mathcal{C}_{i,1}, \ldots, \mathcal{C}_{i,m}$, which do not overlap (except at their endpoints).*

Before proceeding on how to obtain $\mathcal{C}_i$, we need the following technical lemma:

Lemma 4.7. *Let U and W be two subsets of points strictly separable from each other by a straight line and let (uw) and $(u'w')$ be the two inner bi-tangents of $conv(U)$ and $conv(W)$, such that u and u' are in U, and that the angle $\measuredangle((uw),(w'u'))$ between the two oriented straight lines (uw) and $(w'u')$ is positive. If $w \neq w'$, for every edge w_iw_{i+1} of $conv(W)$ between w and w' in counter clockwise direction, $U \subset (w_i\mathring{w}_{i+1})^-$.*

Proof. (i) Since no three points are collinear, all the points of W belong to $(uw)^-$ and $(u'w')^+$, and all the points of U belong to $(uw)^+$ and $(u'w')^-$. The intersection point c of (uw) and $(u'w')$ belongs then to the line segments uw and $u'w'$. Let $(w_1, w_2, \ldots, w_m)$ be the polygonal line extracted from $\delta(conv(W))$ between the points $w_1 = w$ and $w_m = w'$ in the counter clockwise direction (see Figure 4.5).

When $m = 2$, c and all the other points of U are then on the right of the oritented straight line $w_1w_2 = ww'$.

(ii) Let us show now, that for $m > 2$, for every $i \in \{2, \ldots, m-1\}$, the point w_i is inside the triangle cw_1w_m. From (i), w_i is on the right of (cw_1) and on the left of (cw_m). Moreover, since $w_1 = w$ and $w_m = w'$ are extremal vertices of $conv(W)$, $w_1w_2, \ldots, w_m$ is the boundary of the convex hull of $conv(W) \cap (w_1w_m)^-$. w_i is then on the right of (w_1w_m). This shows that w_i is inside cw_1w_m.

(iii) For all $i \in \{2, \ldots, m-2\}$, the edge (w_iw_{i+1}) of $conv(W)$ is such that w_1 and w_m are on the left of (w_iw_{i+1}). It results from (ii) that the straight line (w_iw_{i+1}) cuts each of the segments cw_1 and cw_m. Moreover, this is also true for $i = 1$ and for $i = m-1$. It results that, for each $i \in \{1, \ldots, m-1\}$, c and the vertices of U are on the right of w_iw_{i+1}. ☐

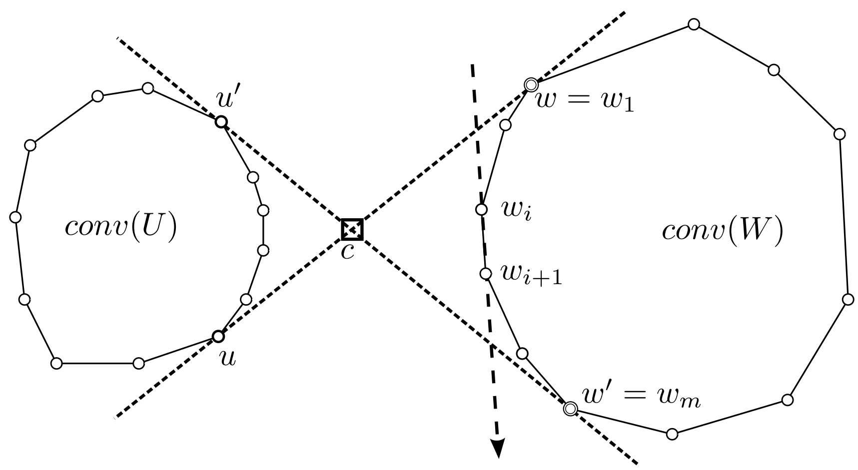

Figure 4.5: The edges w_iw_{i+1} have the points of U on their right side.

Lemma 4.8. *Let $|V_{i-1}| > k$ and let $e_{P_j}(s_j, t_j)$ and $e_{P_{j+1}}(s_{j+1}, t_{j+1})$ be the edges of $g^k(V_{i-1})$ that respectively enters and leaves the vertex $g(T_j)$ of $\mathcal{D}_i$.*

(i) The first and the last vertices of $\mathcal{C}_{i,j}$ are the respective images by an homothety $\mathscr{H}_j$ of center $g(T_j \cup \{v_i\})$ and of ratio $-1/k$ of the following two vertices of $conv(T_j \cup \{v_i\})$:

- *v_i and s_{j+1} when $j = 1$,*
- *t_j and s_{j+1} when $j \in \{2, \ldots, m-1\}$,*
- *t_j and v_i when $j = m$.*

(ii) If $\mathcal{C}_{i,j}$ is not reduced to a unique point, $\mathcal{C}_{i,j}$ is the image by $\mathscr{H}_j$ of the part of $\delta(conv(T_j \cup \{v_i\}))$ that links these two vertices in counter clockwise direction.

Proof. (i.1) Since, by the definition of a convex inclusion chain, v_i is separable from V_{i-1} by a straight line, v_i is a vertex of $conv(T_j \cup \{v_i\})$, for all $j \in \{1, \ldots, m\}$. Now, v_i is also the image of $g(T_1)$ by the homothety $\mathscr{H}_1^{-1}$ of center $g(T_1 \cup \{v_i\})$ and of ratio $-k$ and is also the image of $g(T_m)$ by the homothety $\mathscr{H}_m^{-1}$.

From Lemma 4.5, v_i is then the image of the first vertex of $\mathcal{C}_{i,1}$ by the homothety $\mathscr{H}_1^{-1}$ and the image of the last vertex of $\mathcal{C}_{i,m}$ by the homothety $\mathscr{H}_m^{-1}$ of center $g(T_m \cup \{v_i\})$ and ratio $-k$.

(i.2) From Proposition 2.12, $T_j = P_j \cup \{t_j\} \subset (s_j t_j)^-$ and, since for every $j \in \{2, \ldots, m\}$ $e_{P_j}(s_j, t_j)$ is not an edge of $g^k(V_i)$, $v_i \in (\overset{\circ}{s_j t_j})^-$ from Lemma 3.6. It results that t_j is a vertex of $conv(T_j \cup \{v_i\})$, for every $j \in \{2, ..., m\}$. Now, t_j is also the image of $g((T_j \cup \{v_i\}) \setminus \{t_j\}) = g(P_j \cup \{v_i\})$ by the homothety $\mathscr{H}_j^{-1}$ of center $g(T_j \cup \{v_i\})$ and ratio $-k$. Moreover, from Lemma 4.5, $g(P_j \cup \{v_i\})$ is the first vertex of $\mathcal{C}_{i,j}$.

(i.3) In the same way, for every $j \in \{1, \ldots, m-1\}$, $T_j = P_{j+1} \cup \{s_{j+1}\} \subset (s_{j+1} t_{j+1})^-$ and, since $e_{P_{j+1}}(s_{j+1}, t_{j+1})$ is not an edge of $g^k(V_i)$, $v_i \in (\overset{\circ}{s_{j+1}, t_{j+1}})^-$. s_{j+1} is then a vertex of $conv(T_j \cup \{v_i\})$, for every $j \in \{1, \ldots, m-1\}$. Now, s_{j+1} is also the image of $g((T_j \cup \{v_i\}) \setminus \{s_{j+1}\}) = g(P_{j+1} \cup \{v_i\})$ by the homothety $\mathscr{H}_j^{-1}$ of center $g(T_j \cup \{v_i\})$ and ratio $-k$. Hence, $g(P_{j+1} \cup \{v_i\})$ is the last vertex of $\mathcal{C}_{i,j}$.

(ii.1) Let us study first the case where $j \in \{2, \ldots, m-1\}$ (see Figure 4.7). $\mathcal{C}_{i,j}$ is not reduced to a single point and from Proposition 3.8 it is not equal to the whole boundary of $g^k(V_i)$. Hence the endpoints of $\mathcal{C}_{i,j}$ are distinct, *i.e.* $g((T_j \setminus \{t_j\}) \cup \{v_i\})$ is distinct from $g((T_j \setminus \{s_{j+1}\}) \cup \{v_i\})$, and so $t_j \neq s_{j+1}$.

Moreover, from (i.2) and (i.3), $T_j \cup \{v_i\} \subset (s_j t_j)^- \cap (s_{j+1} t_{j+1})^-$ and, from Proposition 2.12, $V_i \setminus T_j \subset (s_j t_j)^+ \cap (s_{j+1} t_{j+1})^+$. $(s_j t_j)$ and $(s_{j+1} t_{j+1})$ are then two inner bi-tangents of $conv(T_j \cup \{v_i\})$ and $conv(V_{i-1} \setminus T_j)$. Furthermore, since

$\{t_j, s_{j+1}\} \subseteq T_j$ and since $\measuredangle((s_jt_j),(s_{j+1}t_{j+1})) > 0$, the edges of $conv(T_j \cup \{v_i\})$ between t_j and s_{j+1} in counter clockwise direction have all the points of $V_{i-1} \setminus T_j$ on their right, from Lemma 4.7. For such an oriented edge qq' of $conv(T_j \cup \{v_i\})$ we then have $(T_j \cup \{v_i\}) \setminus \{q, q'\} \subset (\mathring{qq'})^+$ and $V_{j-1} \setminus T_j \subset (\mathring{qq'})^-$. From Proposition 2.12, it results that $e_{(T_j \cup \{v_i\})\setminus\{q,q'\}}(q', q)$ is an edge of $g^k(V_i)$ and this edge belongs to $\mathcal{C}_i$, from Lemma 3.6. Moreover, $0 < \measuredangle((s_jt_j),(q'q)) < \measuredangle((s_jt_j),(s_{j+1}t_{j+1}))$ and since the edge $e_{(T_j \cup \{v_i\})\setminus\{q',q\}}(q', q)$ is parallel to $(q'q)$, it results that this edge links two vertices of $\mathcal{C}_{i,j}$ and is thus an edge of $\mathcal{C}_{i,j}$. Since $e_{(T_j \cup \{v_i\})\setminus\{q,q'\}}(q', q)$ is the image of $q'q$ by the homothety $\mathscr{H}_j$, it follows that the image by $\mathscr{H}_j^{-1}$ of the part of $\delta(conv(T_j \cup \{v_i\}))$ that links t_{i-1} to s_j in counter clockwise direction, is a subset of a polygonal line included in $\mathcal{C}_{i,j}$. Now, from (i.2) and (i.3), this polygonal line links both endpoints of $\mathcal{C}_{i,j}$ and thus is equal to $\mathcal{C}_{i,j}$.

(ii.2) Let us discuss now, the case where $j = 1$ (see Figure 4.6). As in (ii.1), from Proposition 2.12, $T_1 \subset (s_1t_1)^- \cap (s_2t_2)^-$ and $V_{i-1} \setminus T_1 \subset (s_1t_1)^+ \cap (s_2t_2)^+$. Since $e_{P_1}(s_1, t_1)$ is an edge of $g^k(V_{i-1})$ and of $g^k(V_i)$, $v_i \in (\mathring{s_1t_1})^+$ from Lemma 3.5. Thus, since $conv(T_1) \subset (s_1t_1)^-$ and $t_1 \in T_1$, t_1 is a vertex of $conv(T_1)$ visible from v_i. Moreover, since $e_{P_2}(s_2, t_2)$ is not an edge of $g^k(V_i)$, $v_i \in (\mathring{s_2t_2})^-$ from Lemma 3.5. Thus, since $conv(T_1) \subset (s_2t_2)^-$ and since $s_2 \in T_1$, s_2 is a vertex of $conv(T_1 \cup \{v_i\})$.

If $t_1 = s_2$, v_is_1 is necessarily an edge of $conv(T_1 \cup \{v_i\})$. Since $v_i \in (\mathring{s_1t_1})^+ \cap (\mathring{s_2t_2})^-$, we have $0 < \measuredangle((s_1t_1),(s_2v_i)) < \measuredangle((s_1t_1),(s_2t_2))$. Thus, $T_1 \cup \{v_i\} \subset (v_is_2)^+$ and $V_{i-1} \setminus T_1 \subset (\mathring{v_is_2})^-$. It results that $e_{T_1\setminus\{s_2\}}(s_2, v_i)$ is an edge of $g^k(V_i)$ that belongs to $\mathcal{C}_i$. From (i.1) and (i.3), its endpoints $g(T_1)$ and $g((T_1 \cup \{v_i\}) \setminus \{s_2\})$ are also the endpoints of $\mathcal{C}_{i,1}$ and are the images of v_i and of s_2 by $\mathscr{H}_1$.

If $t_1 \neq s_2$, $conv(T_1 \cup \{v_i\})$ admits necessarily an edge that links v_i to a vertex q of $conv(T_1)$ that is between t_1 and s_2 in counter clockwise direction (t_1 and s_2 included). Moreover, $conv(T_1 \cup \{v_i\}) \subset (v_iq)^+$. Since t_1 and s_2 belong to T_1 and since $v_i \in (\mathring{s_1t_1})^+ \cap (\mathring{s_2t_2})^-$, it results that $0 < \measuredangle((s_1t_1),(qv_i)) < \measuredangle((s_1t_1),(s_2t_2))$. Thus, $V_{i-1} \setminus T_1 \subset (\mathring{v_iq})^-$ and $e_{T\setminus\{q\}}(q, v_i)$ is an edge of $\mathcal{C}_i$. This edge is furthermore the image by $\mathscr{H}_1$ of v_iq. Every other edge $q'q''$ of $conv(T_1 \cup \{v_i\})$ bewteen v_i and s_2 in the counter clockwise direction is also an edge of $conv(T_1)$ and is between the vertices t_1 and s_2 in counter clockwise direction. Now, since $T_1 \subset (s_1t_1)^- \cap (s_2t_2)^-$ and since $V_{i-1} \setminus T_1 \subset (s_1t_1)^+ \cap (s_2t_2)^+$, (s_1t_1) and (s_2t_2) are the inner bi-tangents of $conv(T_j)$ and $conv(V_{i-1} \setminus T_1)$. Thus, the edges of $conv(T_1)$ between t_1 and s_2 in counter clockwise direction, have all the points of $V_i \setminus T_1$ on their right, from Lemma 4.7. Hence, $(T_1 \cup \{v_i\}) \setminus \{q', q''\} \subset (\mathring{q'q''})^+$ and $V_{i-1} \setminus T_1 \subset (\mathring{q'q''})^-$. $e_{(T_1 \cup \{v_i\})\setminus\{q',q''\}}(q'', q')$ is then an edge of $g^k(V_i)$ and, as in (ii.1), this edge belongs to $\mathcal{C}_{i,1}$ and is an image of $q''q'$ by the homothety $\mathscr{H}_1$. It results that the image by $\mathscr{H}_1$ of the part of $\delta(conv(T_1 \cup \{v_i\}))$ that links v_i to s_2 in counter clockwise direction is a polygonal line within $\mathcal{C}_{i,1}$. Moreover, from (i.1) and (i.3), this polygonal line links the endpoints of $\mathcal{C}_{i,1}$ and is thus equal to $\mathcal{C}_{i,1}$.

(ii.3) A symmetric proof, shows that, when $j = m$, $\mathcal{C}_{i,m}$ is the image by $\mathscr{H}_m$ of the part of $\delta(conv(T_m \cup \{v_i\}))$ between t_m and v_i in counter clockwise direction (see Figure 4.8). $\square$

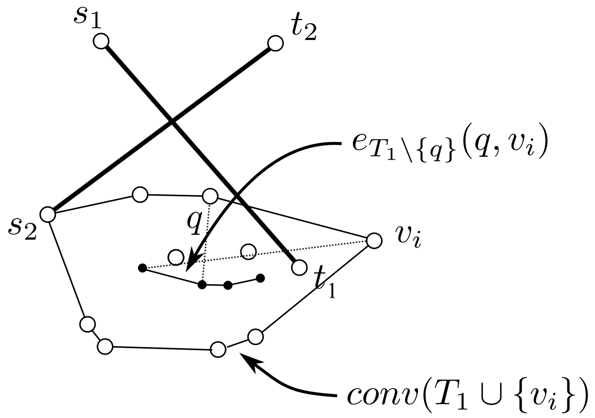

Figure 4.6: Building $e_{T_1\setminus\{q\}}(q, v_i)$

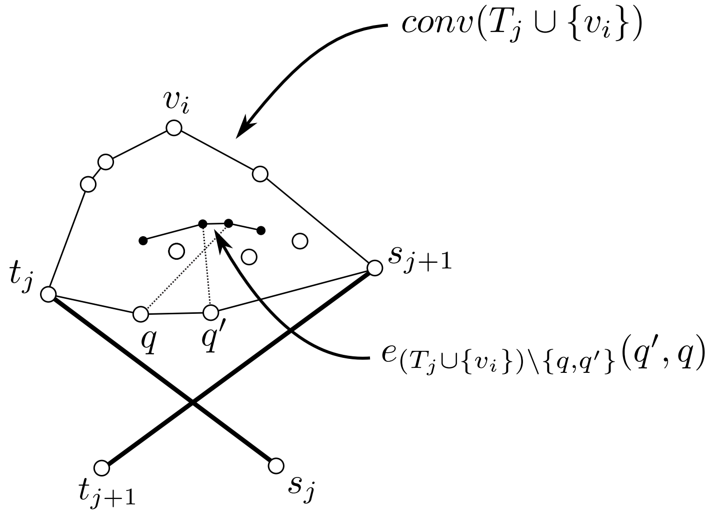

Figure 4.7: Building $e_{(T_j\cup\{v_i\})\setminus\{q,q'\}}(q', q)$

Remark 4.9. *Since $e_{P_j}(s_j, t_j)$ and $e_{P_{j+1}}(s_{j+1}, t_{j+1})$ are edges of $g^k(V_{i-1})$, then $s_{j+1} \neq v_i$ and $t_j \neq v_i$. It then results from Lemma 4.8 that $\mathcal{C}_{i,1}$ and $\mathcal{C}_{i,m}$ are not reduced to points. Now, from Proposition 3.8, the first edge and the last edge of $\mathcal{C}_i$ are of the form $e_P(s, v_i)$ and $e_{P'}(v_i, t')$ respectively. Hence, these two edges are also the first edge of $\mathcal{C}_{i,1}$ and the last edge of $\mathcal{C}_{i,m}$ respectively. All the other edges of $\mathcal{C}_{i,1}$ and $\mathcal{C}_{i,m}$ and all the edges of $\mathcal{C}_{i,j}$, with $j \in \{2, \ldots, m-1\}$ are of the form $e_{P''}(s'', t'')$, with $v_i \in P''$.*

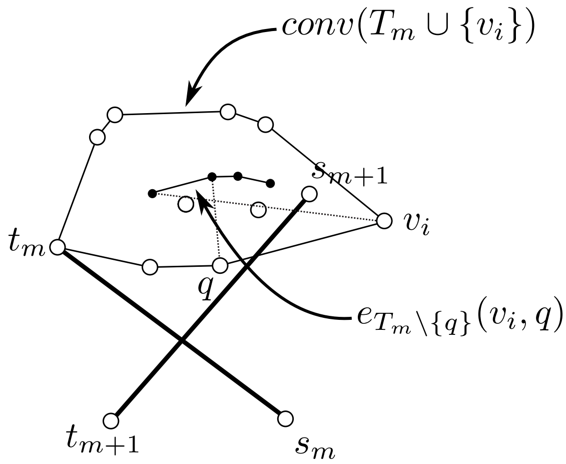

Figure 4.8: Building $e_{T_m \setminus \{q\}}(v_i, q)$

4.4 Incremental construction of k-sets of a special convex inclusion chain

The results in the previous section are going to be used now to develop an incremental construction algorithm of the k-set polygon of V. Actually, we will be interested in the case where all the points of V are ordered as a convex inclusion chain that is also a simple polygonal line.

4.4.1 Data structure

First the data structure to implement is described.

From Proposition 2.12, for every set S, two k-sets of S whose centers of gravity define an edge of $g^k(S)$ differ from each other by one site. Assuming that we know a k-set T of S and its center of gravity $g(T)$, we get to the next vertex $g(T')$ of $g^k(S)$ by simply replacing a point s of T by a point t to get T'. Thus, it suffices to know one k-set T, and the points s and t for each edge to be able to generate all the k-sets of S while traversing the boundary of $g^k(S)$. Every edge $e_P(s,t)$ of $g^k(S)$ can then be represented by the following data structure:

```
structure edge
{
  s, t;    // The points s and t that together with P
       // characterize e_P(s,t).

  next, prev; // The next and previous edges of e_P(s,t)
       // on δ(g^k(S))
}
```

Lemma 4.8 proves that our algorithm will not only need the set of k-sets but also their convex hulls. Moreover, we need to add and remove points from these convex hulls while traversing the boundary of the k-set polygon.

To this purpose, we employ a data structure that allows dynamic convex hull maintenance. Using results given by Overmars and van Leeuwen [OvL81] (see also Overmars [Ove83]), this structure needs $O(h)$ size to store the convex hull of h points of the plane and allows to get the predecessor and the successor of any edge in constant time. It can also be updated in $O(\log^2 h)$ time after inserting or deleting a point. Thus the data structure to store the k-set polygon.

```
structure k_set_polygon
{
  CH; // The convex hull of a k-set T stored in a dynamic
     // convex hull data structure

  e; // The edge of the k-set polygon entering in g(T).
     // This edge serves as first edge in the edge list.
}
```

Theorem 4.10. *A k-set polygon with c edges can be stored in a data structure of size $O(c + k)$.*

Proof. The dynamic convex hull data structure of Overmars and van Leeuwen [OvL81], needs only an $O(k)$ space to store the convex hull of k points. Moreover, the edge list requires $O(c)$ space to store the c edges of the k-set polygon, thus the total space needed to store the k-set polygon is in $O(c + k)$. □

4.4.2 Constructing $g^k(V_{k+1})$

Let $(v_1, \ldots, v_n)$ be a convex inclusion chain of V that forms a simple polygonal line. For all $i \in \{1, \ldots, n\}$, let $V_i = \{v_1, \ldots, v_i\}$. In the following, the k-set polygon of V_i, will also be called the k-set polygon of $(v_1, \ldots, v_n)$.

Since the k-set polygon of a set of k-points is reduced to their center of gravity, the first case where we need to construct a k-set polygon, is when we have a set of $k + 1$ points.

Lemma 4.11. *ts is a counter-clockwise oriented edge of $conv(V_{k+1})$ if and only if $e_{V_{k+1}\setminus\{s,t\}}(s,t)$ is an edge of $g^k(V_{k+1})$. Moreover, if $e_{V_{k+1}\setminus\{s',t'\}}(s',t')$ is the successor of $e_{V_{k+1}\setminus\{s,t\}}(s,t)$ on $\delta(g^k(V_{k+1}))$, $t's'$ is the successor of ts on $\delta(conv(V_{k+1}))$.*

Proof. If ts is an edge of $conv(V_{k+1})$, all the points of $V_{k+1} \setminus \{t, s\}$ belong to $(st)^-$. From Proposition 2.12, $e_{V_{k+1}\setminus\{s,t\}}(s,t)$ is then an edge of $g^k(V_{k+1})$. Conversely, from this same proposition, all the edges of $g^k(V_{k+1})$ are of this form. In addition, if $e_{V_{k+1}\setminus\{s',t'\}}(s',t')$ is the successor of $e_{V_{k+1}\setminus\{s,t\}}(s,t)$ on $\delta(g^k(V_{k+1}))$, then $(V_{k+1} \setminus \{s,t\}) \cup \{t\} = (V_{k+1} \setminus \{s',t'\}) \cup \{s'\}$, that is, $s = t'$. ts and $t's'$ are therefore two consecutive edges of $conv(V_{k+1})$. $\square$

It follows that constructing $g^k(V_{k+1})$ comes to construct $conv(V_{k+1})$ and this can be done with the algorithm described in Section 4.2. We also initialize the dynamic convex hull data structure CH with the convex hull of the k-set V_k. The first edge e is thus the edge $e_{V_{k+1}\setminus\{v_{k+1},t\}}(v_{k+1},t)$ that enters in $g(V_k)$. Thus the algorithm that initializes the k-set polygon data structure with $g^k(V_{k+1})$:

```
function k_set_polygon::initialize(v1,...,vk+1)
{
  this = construct_convex_hull(v1,...,vk+1);
  // The k_set_polygon data structure is initialized
  // with the convex hull of Vk+1, the field e
  // beeing initialized with the edge of this
  // convex hull entering in its rightmost vertex vk+1

  Let e' = e;

  do {
    swap(e'.s, e'.t);
    e' = e'.next;
  } while( e' ≠ e );

  // e is now the edge entering in the leftmost
  // vertex g(Vk) of g^k(Vk+1)

  CH = dynamic_convex_hull(Vk);
}
```

Proposition 4.12. *The previous algorithm builds $g^k(V_{k+1})$ in $O(k\log^2 k)$ time.*

Proof. From Theorem 4.2, the function `construct_convex_hull` takes $O(k)$ time to build the convex hull of V_{k+1}. Next, the do-while loop traverses the at most $k+1$ edges of $conv(V_{k+1})$. Finally, the dynamic convex hull data structure of the k-points is constructed in $O(k\log^2 k)$ time [OvL81]. Thus, the whole previous algorithm runs in $O(k\log^2 k)$ time. $\square$

4.4.3 Finding an extremum of $\mathcal{D}_i$

Using Lemma 3.5, we can easily find the polygonal line of the edges to remove once we add v_i, provided that we have at least one edge e of $\mathcal{D}_i$.

Lemma 4.13. *At least one edge of $\mathcal{C}_{i-1}$ is an edge of $\mathcal{D}_i$.*

Proof. Since $(v_1, \ldots, v_i)$ is a convex inclusion chain that is also a simple polygonal line, v_{i-1} is a vertex of $conv(V_{i-1})$ and from Lemma 4.1, v_{i-1} is also visible from v_i. Thus, there exists an oriented straight line Δ passing through v_{i-1}, that is not parallel to any straight line passing through any two points of V_{i-1}, and such that $conv(V_{i-1}) \subset \Delta^+$ and $v_i \in \mathring{\Delta}^-$ (see Figure 4.9). Let Δ' be a straight line parallel to Δ, oriented in the same direction as Δ and such that $|\mathring{\Delta'}^- \cap V_{i-1}| = k$. Let $V' = \mathring{\Delta'}^- \cap V_{i-1}$. Let (st) and $(s't')$ be the oriented straight lines tangent to both $conv(V')$ and $conv(V_{i-1} \setminus V')$ such that $\{s', t\} \subseteq V'$, $conv(V') \subset (st)^-$, and $conv(V_{i-1} \setminus V') \subset (st)^+$ (resp. $conv(V') \subset (s't')^-$, and $conv(V_{i-1} \setminus V') \subset (s't')^+$). Thus, from Proposition 2.12 and Lemma 3.6, $e_{V' \setminus \{t\}}(s,t)$ and $e_{V' \setminus \{s'\}}(s',t')$ are edges of $g^k(V_{i-1})$ that belong to $\mathcal{C}_{i-1}$, since $v_{i-1} \in V'$.

Since $\mathring{\Delta}'^+$ contains s and t' and $\mathring{\Delta}'^-$ contains s' and t, Δ' cuts all the straight line segments st, $s't'$, ss' and tt'. Let p and p' be the respective intersection points of Δ' with st and $s't'$ and let c be the intersection point of st with $s't'$. Since Δ' cuts the edge ss' of the triangle css', it cuts exactly one of the edges cs and cs' or it contains c. Similarly Δ' cuts exactly one of the edges ct and ct' of the triangle ctt' or it contains c.

Assume that both edges $e_{V' \setminus \{t\}}(s,t)$ and $e_{V' \setminus \{s'\}}(s',t')$ belong to $g^k(V_i)$. Thus, by Lemma 3.5, v_i belongs to $(\mathring{st})^+ \cap (\mathring{s't'})^+$ and, since v_i belongs also to $\mathring{\Delta}'^-$, $(\mathring{st})^+ \cap (\mathring{s't'})^+ \cap \mathring{\Delta}'^- \neq \emptyset$. It follows that c belongs to $\mathring{\Delta}'^-$ and that the points p and p' belong respectively to the straight line segments ct and ct'. Thus $(\mathring{st})^+ \cap (\mathring{s't'})^+ \cap \mathring{\Delta}'^-$ is contained in the triangle cpp'. But $conv(V_{i-1})$ contains the points c, p and p' and hence the triangle cpp' but this is impossible since v_i does not belong to $conv(V_{i-1})$. This proves that at least one of the edges $e_{V' \setminus \{t\}}(s,t)$ and $e_{V' \setminus \{s'\}}(s',t')$ of $\mathcal{C}_{i-1}$ belongs to $\mathcal{D}_i$.

$\square$

This lemma shows that it suffices to know one edge of $\mathcal{C}_{i-1}$ to find and get an edge of $\mathcal{D}_i$ by only traversing the edges of $\mathcal{C}_{i-1}$. Similarly to what we have done in the previous subsection, after the insertion of every point v_{i-1}, we store in the first edge e of the current k-set polygon data structure the unique edge of the polygonal line $\mathcal{C}_{i-1}$ that is of the form $e_P(v_{i-1}, t)$. The data structure CH contains then the convex hull of $P \cup \{t\}$. Note that, in the case where

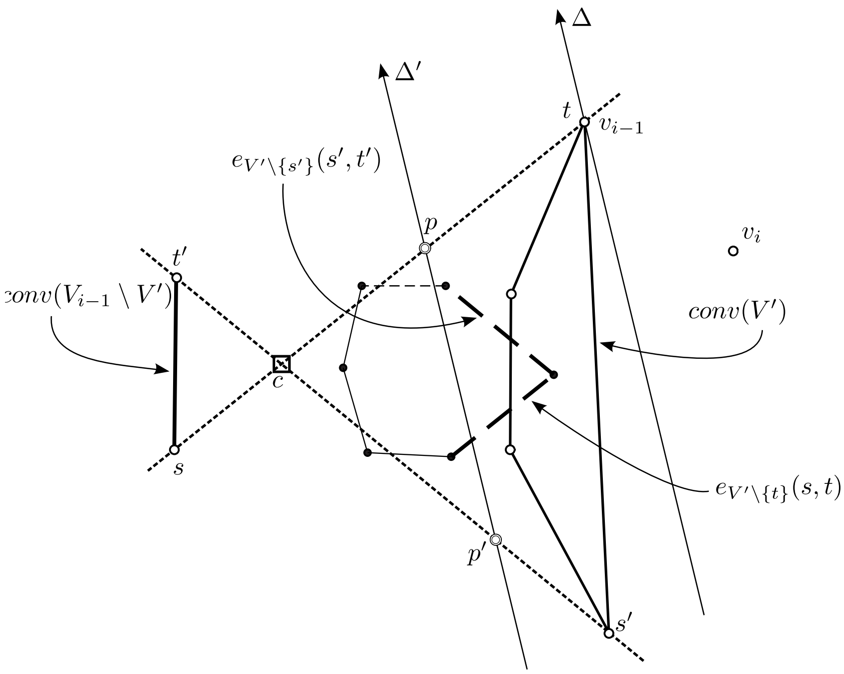

Figure 4.9: Illustration of the proof of Lemma 4.13 in the case where $k = 4$

$i-1 \geq k+2$, $e_P(v_{i-1}, t)$ and $g(P \cup \{t\})$ are the last edge and the last vertex of $\mathcal{C}_{i-1}$ in the clockwise direction.

Let us suppose that for an integer $i \in \{k+2, ..., n-1\}$, we have a k-set polygon data structure initialized in this way and that represents the boundary of $g^k(V_{i-1})$. To find an edge of $\mathcal{D}_i$, when inserting the point v_i, it suffices then to use the results of Lemma 3.5 which states that the edges of $\mathcal{D}_i$ are the edges of $g^k(V_{i-1})$ of the form $e_{P'}(s', t')$ with $v_i \in (\mathring{s't'})^-$.

This leads to the following algorithm that stores in CH the convex hull of T_1, where $g(T_1)$ is the first vertex of $\mathcal{D}_i$ in the clockwise direction. The algorithm also stores in e the edge of $g^k(V_{i-1})$ entering in $g(T_1)$, *i.e.* the last edge of $g^k(V_{i-1})$ (in the counter clockwise direction) that has not been removed.

```
function k_set_polygon::find_conv_t1(v_i)
{
  // find an edge of D_i
  while( v_i is on the left of (e.s, e.t) )
  {
    CH.remove(e.t);
    CH.add(e.s);
    e = e.prev;
  }
```

```
  // find conv(T_1)
  while( v_i is on the right of (e.s, e.t) )
  {
    CH.remove(e.t);
    CH.add(e.s);
    e = e.prev;
  }
}
```

Denote now by $|\mathcal{L}|$ the number of edges of any polygonal line $\mathcal{L}$.

Lemma 4.14. *The function* `find_conv_t1` *runs in* $O((|\mathcal{C}_{i-1}| + |\mathcal{D}_i|)log^2k)$ *time.*

Proof. Since e is initialized with the last edge of $\mathcal{C}_{i-1}$, from Lemma 4.13, the first loop visits at most all the edges of $\mathcal{C}_{i-1}$. The second loop visits at most all the edges of $\mathcal{D}_i$.

Moreover, CH always contains the convex hull of k or $k-1$ points. Thus the insertion or removal of a point in CH is in $O(\log^2 k)$. It results that the total complexity of the function is in $O((|\mathcal{C}_{i-1}| + |\mathcal{D}_i|)log^2k)$ time. □

4.4.4 Constructing $\mathcal{C}_i$

From Corollary 4.6, building the polygonal line $\mathcal{C}_i$ comes down to building the polygonal lines $\mathcal{C}_{i,1}, ..., \mathcal{C}_{i,m}$ and to link them together in this order. From Lemma 4.8, each polygonal line $\mathcal{C}_{i,j}$, can be obtained by extracting a boundary part of the convex hull of $T_j \cup \{v_i\}$. Thus, once we have the k-set polygon of V_{i-1} and v_i, the following algorithm can build the k-set polygon of V_i:

```
function k_set_polygon::build_ci(v_i)
{
  // building conv(T_1) in CH and storing in e the edge of
      g^k(V_{i-1}) entering in g(T_1)
  find_conv_t1(v_i);

  CH.add(v_i);

  Let e_D = e.next; // first edge e_{P_2}(s_2, t_2) of D_i

  // construction of C_{i,1}
  Let s_{j+1} = e_D.s;
  1. foreach( edge qq' between v_i and s_{j+1} on CH in counter
      clockwise direction )
```

```
{
  Let e_new be a new edge;
  e_new.s = q';
  e_new.t = q;
  e_new.prev = e;
  e_D.next = e_new;
  e_D = e_new;
}

CH.remove(s_{j+1});
Let t_j = e_D.t;
CH.add(t_j); // CH = conv(T_j ∪ {v_i})
e_D = e_D.next; // e_D = e_{P_{j+1}}(s_{j+1}, t_{j+1})

// construction of C_{i,j}, for all j ∈ {2,...,m−1}, i.e. while
   e_{P_{j+1}}(s_{j+1}, t_{j+1}) is an edge of D_i
2. while( v_i is on the right of (e_D.s, e_D.t) )
{
  s_{j+1} = e_D.s;
  if( s_{j+1} ≠ t_j )
    3. foreach( edge qq' between t_j and s_{j+1} on CH in counter
       clockwise direction )
    {
      Let e_new be a new edge;
      e_new.s = q';
      e_new.t = q;
      e_new.prev = e;
      e_D.next = e_new;
      e_D = e_new;
    }

  CH.remove(s_{j+1});
  t_j = e_D.t;
  CH.add(t_j);
  e_D = e_D.next;
}

// construction of C_{i,m}
4. foreach( edge qq' between t_j and v_i on CH in counter
   clockwise direction )
{
```

```
    Let e_new be a new edge;
    e_new.s = q';
    e_new.t = q;
    e_new.prev = e;
    e_D.next = e_new;
    e_D = e_new;
  }

  e.next = e_D;
  e_D.prev = e;

  CH.remove(v_i); // CH = conv(T_m)
}
```

Notice that at the end of this algorithm CH contains the convex hull of T_m, with $g(T_m)$ the last point of $\mathcal{C}_i$. Moreover, e is the edge of $\mathcal{C}_i$ entering $g(T_m)$, i.e., the last edge of $\mathcal{C}_i$.

Proposition 4.15. *The previous function builds $\mathcal{C}_i$ in $O((|\mathcal{C}_{i-1}| + |\mathcal{D}_i|) \log^2 k + |\mathcal{C}_i|)$ time.*

Proof. From Lemma 4.14, the complexity of `find_conv_t1` is in $O((|\mathcal{C}_{i-1}| + |\mathcal{D}_i|) \log^2 k)$.

Let us consider now the other instructions of the function. The number of insertions and removals in CH is equal (within a margin of one) to the number of visits in the loop 2. Since each insertion and removal is done in a convex hull of k or $k + 1$ points, the total number of insertions and removals in CH is in $O(|\mathcal{D}_i| \log^2 k)$.

On each visit in one of the foreach loops 1, 3 and 4, a new edge of $\mathcal{C}_i$ is created. The number of other instructions that we run in each of these loops is constant. It results that the total complexity of the foreach loops 1, 3 and 4 is in $O(|\mathcal{C}_i|)$.

The total complexity of the function is then in $O((|\mathcal{C}_{i-1}| + |\mathcal{D}_i|) \log^2 k + |\mathcal{C}_i|)$ time. □

4.4.5 On-line algorithm

An on-line construction algorithm of the k-set polygon of the convex inclusion chain $(v_1, \ldots, v_n)$ that is also a simple polygonal line requires the construction of the k-set polygon of V_{k+1} then adding the points $v_{k+2}, ..., v_n$ one by one and updating the k-set polygon after each insertion.

```
function k_set_polygon::build(v_1,...,v_n)
{
  initialize(v_1,...,v_{k+1});

  for( i = k+2 to n )
    build_ci(v_i);
}
```

Theorem 4.16. *The k-set polygon of a convex inclusion chain which is a simple polygonal line can be constructed on-line in $O(k(n-k)\log^2 k)$ time.*

Proof. The k-set polygon of V_{k+1} is constructed in $O(k\log^2 k)$ time from Proposition 4.12. From Proposition 4.15, $g^k(V_i)$ is computed from $g^k(V_{i-1})$ in $O((|\mathcal{D}_i| + |\mathcal{C}_{i-1}|)\log^2 k + |\mathcal{C}_i|)$ time, for every $i \in \{k+2, ..., n\}$.

It follows that the k-set polygon of V_n is constructed on-line in time

$$O(k\log^2 k + \sum_{i=k+2}^{n} ((|\mathcal{D}_i| + |\mathcal{C}_{i-1}|)\log^2 k + |\mathcal{C}_i|)$$

As in Chapter 3, $c^k = \sum_{i=k+1}^{n} |\mathcal{C}_i|$, so we have

$$\sum_{i=k+2}^{n} ((|\mathcal{D}_i| + |\mathcal{C}_{i-1}|)\log^2 k + |\mathcal{C}_i|) \leq 2c^k \log^2 k + c^k$$

since the total number $\sum_{i=k+2}^{n} |\mathcal{D}_i|$ of removed edges is less than the total number c^k of created edges. From Proposition 3.9, c^k is in $O(k(n-k))$. Thus the time complexity of the algorithm is $O(k(n-k)\log^2 k)$. □

From Corollary 3.10, any algorithm that incrementally constructs the k-set polygon of a set of n points, in such a way that every newly inserted point is outside of the convex hull of the previously inserted ones, has to generate $\Omega(k(n-k))$ edges. It follows that the time complexity of our on-line algorithm is $O(\log^2 k)$ per edge that has to be created. In the introduction we have seen that the algorithm of Cole, Sharir and Yap [CSY87] finds the set of k-sets of n points in the plane in $O(n\log n + c\log^2 k)$ time, where c is the number of k-sets of these n points. The $n\log n$ factor comes from a preprocessing step that computes the convex hull of n points. From theorem 4.2, this factor can be brought down to n in the case of a convex inclusion chain that is also a simple polygonal line. Thus, the algorithm of Cole, Sharir and Yap can build the k-set polygon of such a chain in $O(c\log^2 k)$ time, where c is the number of edges of the final k-set polygon. The complexity of our algorithm per created edge is then the same as the one of the algorithm of Cole, Sharir and Yap. However, their algorithm can only be used when the whole point set is known in advance. In this case their algorithm is more interesting than ours as it creates less edges.

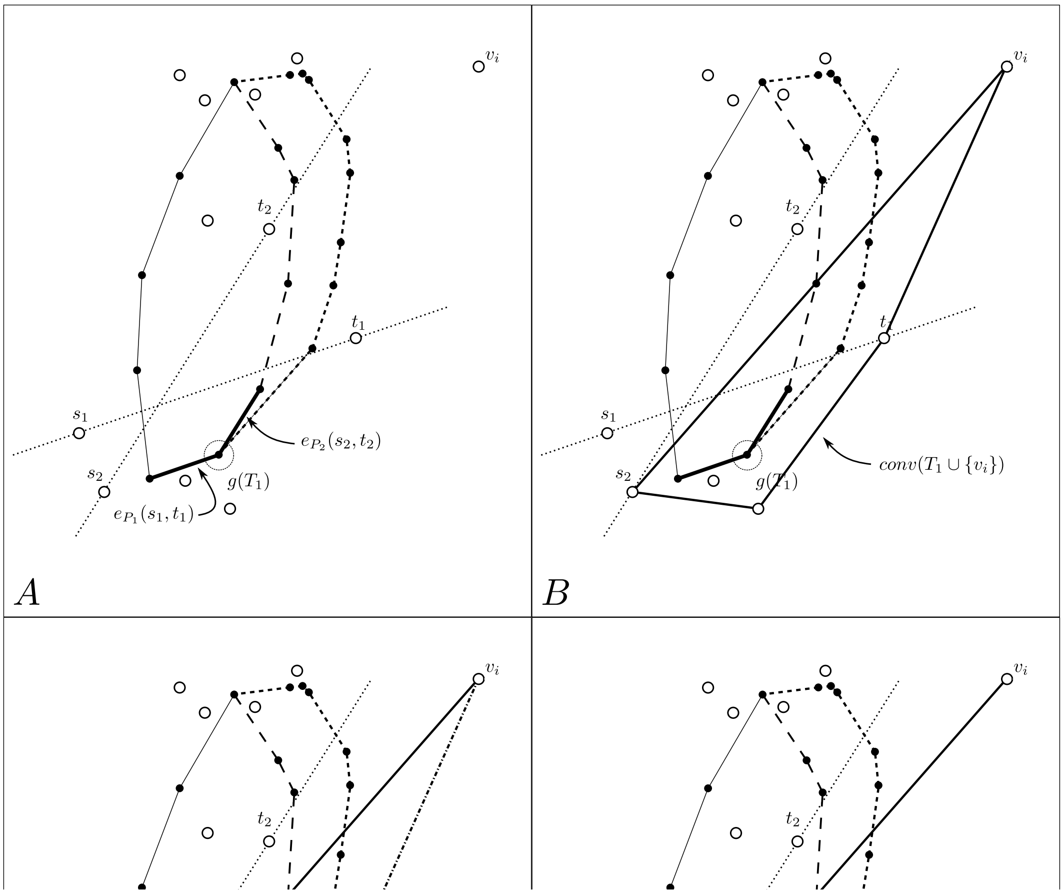

Figure 4.10: Illustration of how the polygonal line part $\mathcal{C}_{i,1}$ gets constructed: In A, we start by taking the edges $e_{P_1}(s_1, t_1)$ and $e_{P_1}(s_2, t_2)$ incident in $g(T_1)$. In B, we build the convex hull of $T_1 \cup \{v_i\}$. In C, we take the edges of $conv(T_1 \cup \{v_i\})$ between v_i and s_2 and finally in D we build their corresponding images by $\mathscr{H}_1$ to get $\mathcal{C}_{i,1}$

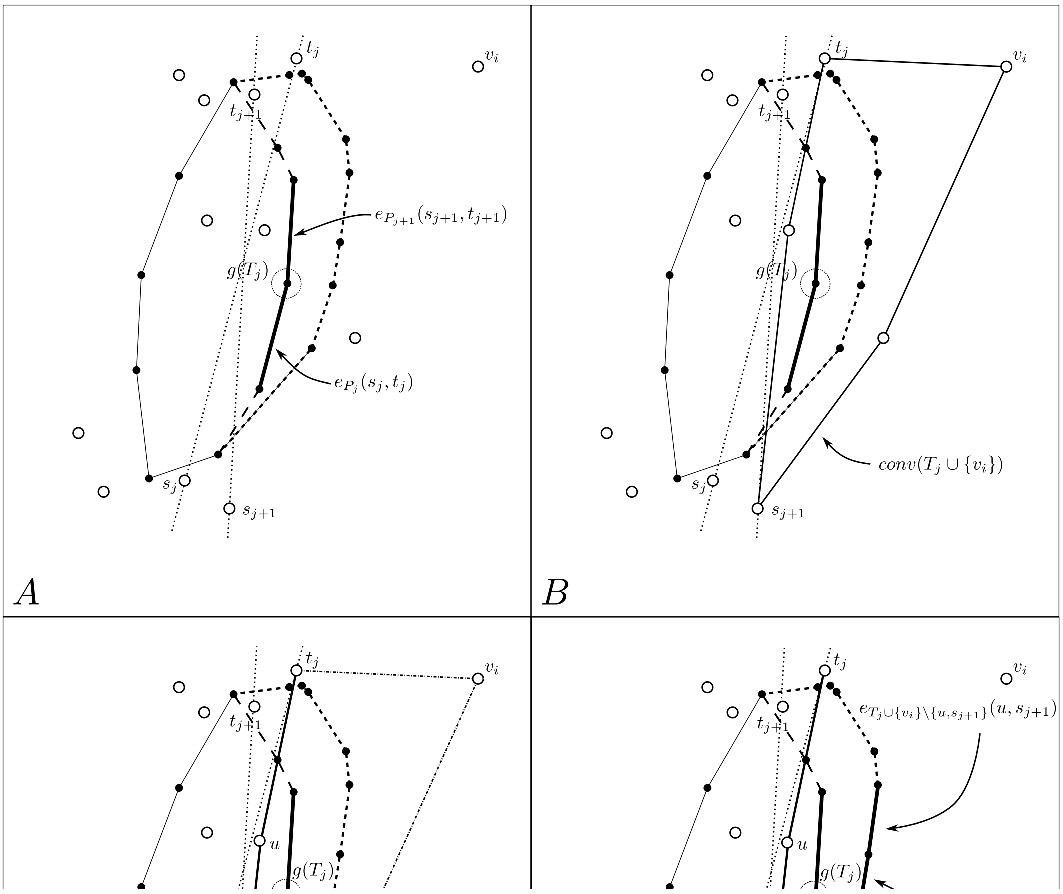

Figure 4.11: Illustration of how the polygonal line part $\mathcal{C}_{i,j}$ gets constructed: In A, we start by taking the edges $e_{P_j}(s_j, t_j)$ and $e_{P_{j+1}}(s_{j+1}, t_{j+1})$ incident in $g(T_j)$. In B, we build the convex hull of $T_j \cup \{v_i\}$. In C, we take the edges $t_j u$ and $u s_{j+1}$ of $conv(T_j \cup \{v_i\})$ between t_j and s_{j+1} and finally in D we build their corresponding images $e_{T_j \cup \{v_i\} \setminus \{t_j, u\}}(t_j, u)$ and $e_{T_j \cup \{v_i\} \setminus \{u, s_{j+1}\}}(u, s_{j+1})$ by $\mathscr{H}_j$ to get $\mathcal{C}_{i,j}$

Chapter 5

Divide and conquer construction of k-set polygons

5.1 Introduction

This chapter extends another classical convex hull construction method to the k-set polygon, namely the divide and conquer method. The algorithm works similarly in that it starts by dividing the set of points V recursively into subsets of relatively equal size, then recursively constructs their k-set polygons, and merges the polygons two by two.

We first recall how the merging works in the case of the convex hulls. The algorithm presented here does not necessarily correspond to the usual way to merge two convex hulls, but it allows introducing the method that will be used later to construct a k-set polygon.

Afterwards, we characterize the merging of two k-set polygons. We show that, as for convex hulls, the edges to remove form two connected polygonal lines. The main difference with the convex hull construction comes from the fact that new vertices have to be created when two k-set polygons are merged. We show that these vertices can be obtained by considering k-set polygons of only $2k$ points.

This leads to an algorithm that constructs the k-set polygon of n points in $O(n \log n + m \log^2 k \log(n/k))$ time, where m is the worst case size of the output.

5.2 Divide and conquer construction of the convex hull

Let $\mathcal{P}$ be a convex polygon. By $\mathcal{P}^{(\cap)}$ we note the edges of $\mathcal{P}$ in the counter clockwise direction going from the rightmost vertex of $\mathcal{P}$ to the leftmost vertex of $\mathcal{P}$, and by $\mathcal{P}^{(\cup)}$ we note the edges of the convex hull in the counter clockwise direction from the leftmost vertex of $\mathcal{P}$ to the rightmost vertex of $\mathcal{P}$ (see Figure 5.1).

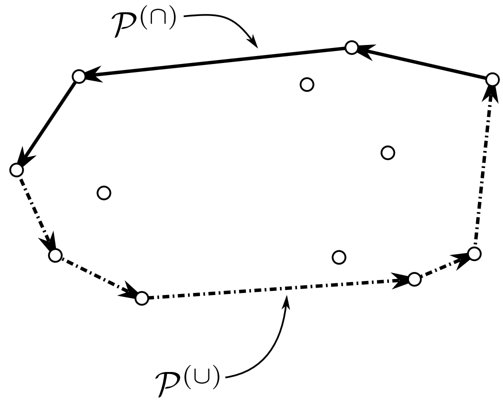

Figure 5.1: The convex polygon $\mathcal{P}$, the upper polygonal line $\mathcal{P}^{(\cap)}$, and the lower polygonal line $\mathcal{P}^{(\cup)}$

It is assumed in the whole chapter that no two points of V belong to a same vertical line (the line can always be chosen in that way). Moreover, for the sake of simplicity of the exposition, we suppose that no four distinct points of V belong to two parallel lines.

We assume that the points of the set V are sorted according to their x-coordinates in lexicographical order. We divide the set V into two different sets V_l and V_r of relatively equal size, such that the x-coordinates of the points of the set V_l are less than the x-coordinates of the points of V_r. Let π be a vertical oriented straight line that separates the set V_l from the set V_r and such that $V_r \subset \mathring{\pi}^-$ and $V_l \subset \mathring{\pi}^+$ (see Figure 5.2).

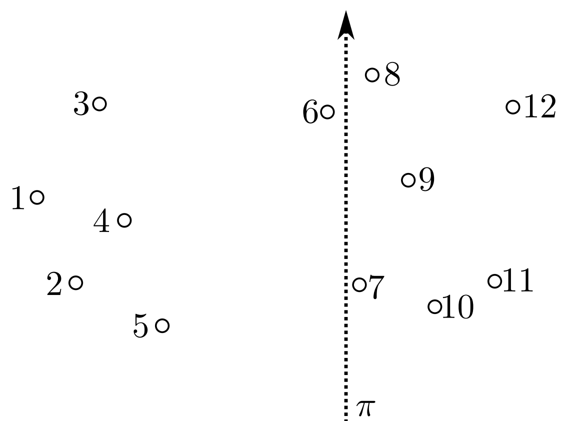

Figure 5.2: The straight line π that separates the set $V_l = \{1, 2, 3, 4, 5, 6\}$ from the set $V_r = \{7, 8, 9, 10, 11, 12\}$

We show now how to build $conv(V = V_r \cup V_l)$ from both $conv(V_r)$ and $conv(V_l)$, assuming that $conv(V_r)$ and $conv(V_l)$ exist.

5.2.1 Edge removal

Similarly to what has been done previously in the case of the incremental construction of $conv(V)$, the edges to remove from both $conv(V_r)$ and $conv(V_l)$ are going to be characterized.

First we prove that:

Lemma 5.1. *(i) If $conv(V_l)$ and $conv(V_r)$ are not reduced to single points, at least one edge incident in the rightmost vertex of $conv(V_l)$ and at least one edge incident in the leftmost vertex of $conv(V_r)$ are not edges of $conv(V_l \cup V_r)$.*

(ii) The leftmost vertex of $conv(V_l)$ and the rigthmost vertex of $conv(V_r)$ are vertices of $conv(V_l \cup V_r)$.

Proof. (i) Let st and $s't'$ be the edges of $conv(V_l)$ incident in the rightmost vertex $t = s'$. Since $conv(V_l) \subset \mathring{\pi}^+$ and $conv(V_r) \subset \mathring{\pi}^-$, it follows that $\mathring{\pi}^- \subset (st)^- \cup (s't')^-$ and $V_r \subset (st)^- \cup (s't')^-$. This means that at least one of the edges st or $s't'$ has a point of V_r on its right, thus at least one of these edges is not an edge of $conv(V_l \cup V_r)$.

Symmetrically, at least one edge incident in the leftmost vertex of $conv(V_r)$ is not an edge of $conv(V_l \cup V_r)$.

(ii) Since the leftmost vertex of $conv(V_l)$ is an extreme point of the set $V_l \cup V_r$, the leftmost vertex can be separated from the set $V_l \cup V_r$ by a straight line. Thus, the leftmost vertex of $conv(V_l)$ is a vertex of $conv(V_l \cup V_r)$. Similarly, the rightmost vertex of $conv(V_r)$ is an extreme point of $V_l \cup V_r$, and thus is a vertex of $conv(V_l \cup V_r)$. □

Now we can characterize which edges to keep, and which ones to remove.

Lemma 5.2. *Let st and $s't'$ be two consecutive edges of $conv(V_l)^{(\cap)}$ in the counter-clockwise direction, and let v_r be a point of V_r.*

If $v_r \in (s't')^-$ then $v_r \in (st)^-$, that is, if $v_r \in (st)^+$ then $v_r \in (s't')^+$.

Proof. Since $(s't')^- \cap \mathring{\pi}^- \subset (st)^- \cap \mathring{\pi}^-$ and since $v_r \in \mathring{\pi}^- \cap (s't')^-$ then $v_r \in (st)^-$ (see Figure 5.3). □

Similar results also hold for $conv(V_r)^{(\cap)}$, $conv(V_l)^{(\cup)}$, and $conv(V_r)^{(\cup)}$. With these results we can finally prove that:

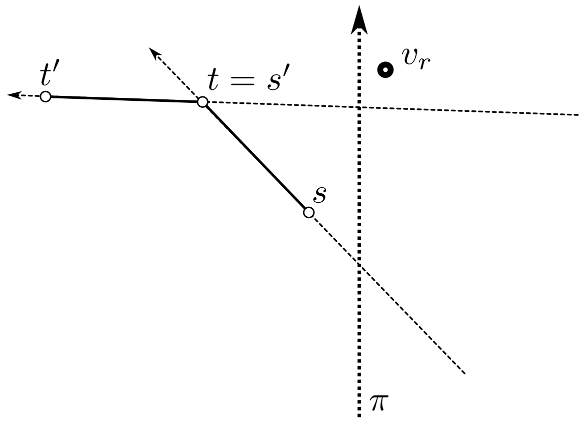

Figure 5.3: Since v_r is on the right of $(s't')$ and of π it is also on the right of (st).

Proposition 5.3. *The edges of $conv(V_l)^{(\cap)}$ (resp. $conv(V_r)^{(\cap)}$) that are not edges of $conv(V_l \cup V_r)$ form a connected polygonal line that contains the rightmost (resp. leftmost) vertex of V_l (resp. V_r).*

Proof. (i) From Lemma 5.2 if an edge of $conv(V_l)^{(\cap)}$ has a point of V_r on its right side, its predecessor edge in counter clockwise direction also has this point on its right side. Recursively, the predecessor of the predecessor also has this point on its right side and so on till we reach the edge incident in the rightmost vertex of $conv(V_l)^{(\cap)}$. This implies that the edges of $conv(V_l)^{(\cap)}$ that are not edges of $conv(V_l \cup V_r)$ form a connected polygonal line starting from the rightmost vertex of $conv(V_l)$.

(ii) Symmetrically, the edges of $conv(V_r)^{(\cap)}$ that are not edges of $conv(V_l \cup V_r)$ form a connected polygonal line that contains the leftmost vertex of $conv(V_r)$. □

Lemma 5.4. *An edge st of $conv(V_l)^{(\cap)}$ is also an edge of $conv(V_l \cup V_r)$ if and only if the vertex v_r of $conv(V_r)$ that is $//_{(st)}$-separable from V_r belongs to $(\mathring{st})^+$.*

Proof. Let Δ be a straight line parallel to st, in the same direction as st and that separates the vertex v_r of $conv(V_r)$ from the rest of the points of V_r.

(i) If $v_r \in (\mathring{st})^+$ then $\Delta \subset (\mathring{st})^+$ and thus, $V_r \subset (\mathring{st})^+$. It follows that st is an edge of $conv(V_r \cup V_l)$ since all the points of V_r are on the left of st.

(ii) Obviously, if $v_r \in (\mathring{st})^-$ then st is not an edge of $conv(V_r \cup V_l)$ since the edge st has a point on its right. □

So now, finding the edges to remove on $conv(V_l)^{(\cap)}$ comes down to finding the first edge of $conv(V_l)^{(\cap)}$ (in counter clockwise direction) that satisfies the previous property.

Obviously, an edge st precedes an edge $s't'$ in counter clockwise direction on $conv(V)^{(\cap)}$ if and only if the angle $\theta(st)$ of the oriented line (st) with the x-axis (oriented from left to right) is smaller than the angle $\theta(s't')$ of $(s't')$ with the x-axis.

The three following notations will be equivalent: $\theta(st) < \theta(s't')$, $(st) <_\theta (s't')$, and $st <_\theta s't'$.

Given a straight line Δ, we can easily find whether a vertex v of $conv(V)$ is $//_\Delta$-separable from V using the following straightforward lemma:

Lemma 5.5. *If the convex hull $conv(V)$ is not reduced to a unique vertex, let $v_0, \ldots, v_m$ be the vertices of $conv(V)^{(\cap)}$, given in counter clockwise direction. Let Δ be an oriented straight line with $\theta(\Delta) \in [\pi/2, 3\pi/2]$. v_i is $//_\Delta$-separable from V if and only if,*
- either $i = 0$ and $\Delta <_\theta (v_0 v_1)$,
- either $i \in \{1, \ldots, m-1\}$ and $(v_{i-1} v_i) <_\theta \Delta <_\theta (v_i v_{i+1})$,
- or $i = m$ and $(v_{m-1} v_m) <_\theta \Delta$.

To avoid dealing with $i = 0$ and $i = m$ as special cases, we only have to add two anchor-edges to $conv(V)^{(\cap)}$. Both edges are vertical, with one ending at the leftmost vertex of $conv(V)^{(\cap)}$ and oriented in the negative y-direction and the other ending at the rightmost vertex of $conv(V)^{(\cap)}$ and oriented in the positive y-direction.

Armed with these results we can easily conceive an algorithm that finds the first edge to keep on $conv(V_l)^{(\cap)}$. We start by testing the edge $s_l t_l$ incident in the rightmost vertex of $conv(V_l)^{(\cap)}$ against the leftmost vertex t of $conv(V_r)^{(\cap)}$. If t is on the left of $(s_l t_l)$, then from Lemma 5.2, it is also on the left of all the successors of $s_l t_l$, so the next vertex to test is the vertex that precedes t on $conv(V_r)^{(\cap)}$. If t is on the right of $(s_l t_l)$ then we advance to the successor of $s_l t_l$ to test it against t. We repeat these steps till we meet the first couple $(s_l t_l, t_r)$ such that t_r is on the left of $(s_l t_l)$ and $s_r t_r <_\theta s_l t_l$, with $s_r t_r$ the edge of $conv(V_r)^{(\cap)}$ ending at t_r. $s_l t_l$ is then the first edge of $conv(V_l)^{(\cap)}$ (in counter clockwise direction) that is not to be removed. Indeed, by construction, every edge preceding $s_l t_l$ is invalidated by a vertex of $conv(V_r)^{(\cap)}$. Moreover, if $s_r t_r <_\theta s_l t_l$, the vertex v_r of $conv(V_r)^{(\cap)}$ that is $//_{(s_l t_l)}$-separable from V_r, follows s_r on $conv(V_r)^{(\cap)}$, from Lemma 5.5. By construction, v_r is then on the left of an edge of $conv(V_l)^{(\cap)}$ preceding t_l. From Lemma 5.2, v_r is also on the left of $(s_l t_l)$ and from Lemma 5.4, $s_l t_l$ has not to be removed. Note that the algorithm stops at the latest when $s_l t_l$ or $s_r t_r$ is an anchor-edge.

Finally the algorithm:

```
function find_upper_left(conv(V_l), conv(V_r))
{
  Let s_l t_l be the edge starting at the rightmost vertex of
      conv(V_l)^(∩);
  Let s_r t_r be the edge ending at the leftmost vertex of
      conv(V_r)^(∩);

  1. while( t_r ∈ (s_l t_l)^- )
  {
    s_l t_l ← successor of s_l t_l on conv(V_l)^(∩);
  }

  2. while( s_l t_l <_θ s_r t_r )
  {
    s_r t_r ← predecessor of s_r t_r on conv(V_r)^(∩);
    3. while( t_r ∈ (s_l t_l)^- )
    {
      s_l t_l ← successor of s_l t_l on conv(V_l)^(∩);
    }
  }

  return s_l t_l;
}
```

Proposition 5.6. *Within a margin of two, only edges to remove are traversed by the algorithm.*

Proof. As already noted, all the edges traversed on $conv(V_l)^{(\cap)}$, except the last one, have to be removed.

Moreover, by construction, for every edge $s_r t_r$ of $conv(V_r)^{(\cap)}$ traversed by the algorithm, except for the last one, there exists an edge $s_l t_l$ of $conv(V_l)^{(\cap)}$ such that $s_l t_l <_\theta s_r t_r$ (loop (2) condition) and $t_r \in (\mathring{s_l t_l})^+$ (loops (1) and (3) conditions). Furthermore, since $t_r \in \mathring{\pi}^-$, $(s_l t_l) \cap \mathring{\pi}^+ \subset (\mathring{s_r t_r})^-$, and since $\{s_l, t_l\} \subset \mathring{\pi}^+$, $\{s_l, t_l\} \subset (\mathring{s_r t_r})^-$. Hence, $s_r t_r$ is not an edge of $conv(V_l \cup V_r)$ and, within a margin of one, only edges to remove are traversed on $conv(V_r)^{(\cap)}$. $\square$

Obviously, the edges to remove on $conv(V_r)^{(\cap)}$, $conv(V_l)^{(\cup)}$, and $conv(V_r)^{(\cup)}$ can be found in the same way.

5.2.2 Edge construction

Now that we know what edges to remove on $conv(V_l)$ and $conv(V_r)$ we can try joining the remaining polygonal lines to form $conv(V_l \cup V_r)$.

In the previous section we found the polygonal lines to keep on both $conv(V_l)$ and $conv(V_r)$. Moreover, we showed that the remaining polygonal line on $conv(V_l)$ is connected and so is the case for the remaining polygonal line on $conv(V_r)$. All the other edges were to remove. Thus, the vertices of the convex hull $conv(V_l \cup V_r)$ are the vertices of the remaining polygonal lines on $conv(V_l)$ and $conv(V_r)$.

Let $l_0, l_1, \dots l_m$ be the set of the vertices on the remaining polygonal line of $conv(V_l)$ in the counter clockwise direction and let $r_0, r_1, \dots r_{m'}$ be the set of the vertices on the remaining polygonal line of $conv(V_r)$ in the counter clockwise direction as well. Since $r_0 \in (l_{m-1}l_m)^+$ and $l_m \in (r_0r_1)^+$, $l_{m-1}l_m <_\theta l_mr_0 <_\theta r_0r_1$ and $conv(V_l) \cup conv(V_r) \subset (l_mr_0)^+$. l_mr_0 is then an edge of $conv(V_l \cup V_r)$. In the same way, $r_{m'}l_0$ is an edge of $conv(V_l \cup V_r)$ (see Figure 5.4). Finally we can deduce the following Lemma:

Lemma 5.7. *Joining the remaining edges on $conv(V_l)$ and $conv(V_r)$ to create $conv(V_l \cup V_r)$ requires the creation of 2 edges only.*

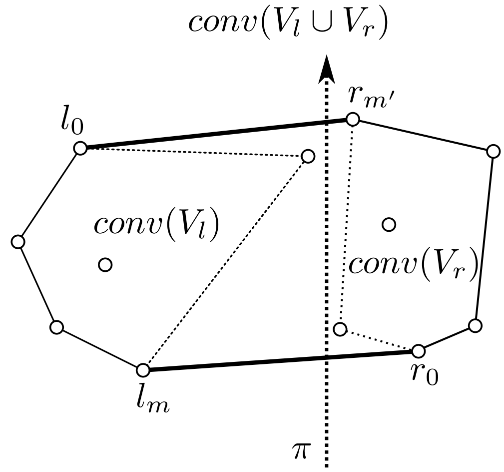

Figure 5.4: Joining $conv(V_l)$ and $conv(V_r)$ to get $conv(V_l \cup V_r)$.

5.2.3 Construction of $conv(V)$

The properties and algorithms given previously can be exploited to create a divide and conquer algorithm to construct the convex hull of a set of points V.

First, the points of V are sorted in a certain lexicographical order (for example according to their x-coordinates). The sorted set V is going to be subdivided into two subsets V_l and V_r of relatively equal size such that the vertices of V_l precede the vertices of V_r in the chosen lexicographical order. The subdivision will continue recursively on V_l and V_r till we have subsets of two or three points. If the size of a subset is two or three we compute the convex hull of this subset otherwise we divide it again into two subsets and then we remove the polygonal lines of the returned convex hulls. Once the remaining polygonal lines are found we can join them and return the resulting convex hull. See the following algorithm for an illustration about the process:

```
function divide_and_conquer(V)
{
  if( |V| ≤ 3 )
    return conv(V);
  else {
    subdivide V into V_l and V_r;
    conv(V_l) = divide_and_conquer(V_l);
    conv(V_r) = divide_and_conquer(V_r);
    upper_left_edge = find_upper_left(conv(V_l), conv(V_r));
    upper_right_edge = find_upper_right(conv(V_l), conv(V_r));
    lower_left_edge = find_lower_left(conv(V_l), conv(V_r));
    lower_right_edge = find_lower_right(conv(V_l), conv(V_r));
    connect the upper_left_edge to the upper_right_edge;
    connect the lower_left_edge to the lower_right_edge;
    return conv(V);
  }
}
```

Theorem 5.8. *Computing the convex hull of a set V of n sorted points using the divide and conquer algorithm takes $O(n)$ time.*

Proof. The only operations performed by the algorithm are creating, deleting, and traversing edges. The total number $Nb(n)$ of created edges verifies the relation:

$$
\begin{aligned}
Nb(n) &= Nb(\lfloor n/2 \rfloor) + Nb(\lceil n/2 \rceil) + 2 && \textit{if } n > 3 \\
Nb(n) &= O(1) && \textit{if } n \leq 3
\end{aligned}
$$

Thus $Nb(n) = O(n)$. The total number of deleted edges is then also bounded by $O(n)$ and, from Proposition 5.6, it is the same with the number of traversed edges. □

5.3 Divide and conquer construction of the k-set polygon

Suppose now that we are given two subsets V_l and V_r of at least k points of V having at most $k-1$ common points and such that there exists a vertical strip containing $V_l \cap V_r$, and that $V_l \setminus V_r$ and $V_r \setminus V_l$ are respectively on the left and on the right side of the strip. More precisely, there exist two vertical straight lines π_l and π_r oriented upwards such that $\pi_l \subset \mathring{\pi}_r^+$, $V_r \subset \mathring{\pi}_l^-$, $V_l \subset \mathring{\pi}_r^+$, and $V_l \cap V_r = (\mathring{\pi}_l^- \cap \mathring{\pi}_r^+) \cap (V_l \cup V_r)$. Suppose furthermore that $g^k(V_l)$ and $g^k(V_r)$ are given (see Figure 5.5). Constructing $g^k(V_l \cup V_r)$ consists, as in the construction of the classical convex hull, in finding the edges to remove on the given k-set polygons $g^k(V_l)$ and $g^k(V_r)$ and afterwards determining the new edges to create in order to construct $g^k(V_l \cup V_r)$.

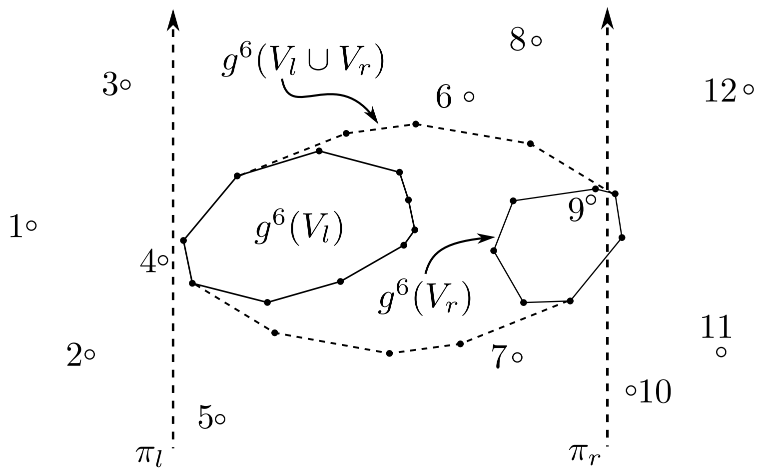

Figure 5.5: The 6-set polygons $g^6(V_l = \{1,2,3,4,5,6,7,8,9\})$, $g^6(V_r = \{5,6,7,8,9,10,11,12\})$, and $g^6(V_l \cup V_r)$.

5.3.1 Edge removal

In this subsection the focus is on the edges to remove. Notably, the characteristics of the two connected lines which they form on $g^k(V_l)$ and $g^k(V_r)$ are given.

Property 5.9. *(i) If $g^k(V_l)$ (resp. $g^k(V_r)$) is not reduced to a unique vertex, at least one of the edges incident in its rightmost (resp. leftmost) vertex is to be removed.*

(ii) The leftmost vertex of $g^k(V_l)$ and the rightmost vertex of $g^k(V_r)$ are vertices of $g^k(V_l \cup V_r)$.

Proof. (i) Since $|V_l \cap V_r| < k$, there exists an oriented vertical straight line Δ such that $g^k(V_l) \subset \mathring{\Delta}^+$ and $g^k(V_r) \subset \mathring{\Delta}^-$. If Δ_1 and Δ_2 are the oriented straight lines generated by the edges incident in the rightmost vertex of $g^k(V_l)$, $\mathring{\Delta}^- \subset \mathring{\Delta}_1^- \cup \mathring{\Delta}_2^-$. Thus, at least one of these edges is not an edge of $g^k(V_l \cup V_r)$. Symmetrically, one of the edges incident in the leftmost vertex of $g^k(V_r)$ is not an edge of $g^k(V_l \cup V_r)$.

(ii) The leftmost vertex of $g^k(V_l)$ is the centroid of the k leftmost points of $V_l \cup V_r$. They can thus be separated from the rest by a vertical straight line and their centroid is a vertex of $g^k(V_l \cup V_r)$, according to Proposition 2.11. In the same way, the rightmost vertex of $g^k(V_r)$ is a vertex of $g^k(V_l \cup V_r)$. □

From now on, $g^k(V)^{(\cap)}$ (resp. $g^k(V)^{(\cup)}$) is considered to be the oriented polygonal line of the edges of $g^k(V)$ that connects the rightmost to the leftmost (resp. leftmost to rightmost) vertex of $g^k(V)$ in counter clockwise direction.

Similarly to the convex hull's simple case, an edge $e_P(s,t)$ precedes an edge $e_{P'}(s',t')$ in counter clockwise direction on $g^k(V)^{(\cap)}$ if, and only if, the edges are such that $e_P(s,t) <_\theta e_{P'}(s',t')$. Thus, since $e_P(s,t)$ and $e_{P'}(s',t')$ are parallel to and oriented in the same directions as st and $s't'$, $e_P(s,t)$ precedes $e_{P'}(s',t')$ if, and only if, $st <_\theta s't'$.

Lemma 5.10. *Let $e_P(s,t)$ and $e_{P'}(s',t')$ be two edges of the line $g^k(V_l)^{(\cap)}$ such that $e_P(s,t) <_\theta e_{P'}(s',t')$ and let r be a point of $V_r \backslash V_l$. If $r \in (\mathring{s't'})^-$ then $r \in (\mathring{st})^-$, that is, if $r \in (\mathring{st})^+$ then $r \in (\mathring{s't'})^+$.*

Proof. If $e_P(s,t)$ and $e_{P'}(s',t')$ are two consecutive edges of $g^k(V_l)^{(\cap)}$, then from Corollary 2.13, the line segments st and $s't'$ intersect and, since $V_l \subset \mathring{\pi}_r^+$, their intersection point belongs to $\mathring{\pi}_r^+$. Moreover, since $st <_\theta s't'$, $(\mathring{s't'})^- \cap \mathring{\pi}_r^- \subset (\mathring{st})^-$. Since every site r of $V_r \setminus V_l$ belongs to $\mathring{\pi}_r^-$, it follows that, if r belongs to $(\mathring{s't'})^-$, it also belongs to $(\mathring{st})^-$ (see Figure 5.6).

By an elementary induction, the result holds for any edges $e_P(s,t)$ and $e_{P'}(s',t')$ of $g^k(V_l)^{(\cap)}$ such that $e_P(s,t) <_\theta e_{P'}(s',t')$. □

Similar results hold for $g^k(V_l)^{(\cup)}$, $g^k(V_r)^{(\cap)}$, and $g^k(V_r)^{(\cup)}$, and thus the following theorem:

Theorem 5.11. *The edges to remove from $g^k(V_l)$ (resp. $g^k(V_r)$) form a connected line which contains the rightmost vertex of $g^k(V_l)$ (resp. leftmost vertex of $g^k(V_r)$).*

Proof. From Proposition 2.12, the edges to remove from $g^k(V_l)$ are the edges $e_P(s,t)$ that have at least one point v of $V_r \setminus V_l$ such that $v \in (\mathring{st})^-$, because the straight line (st) has more than $k-1$ points on its right.

Thus, from Lemma 5.10, if an edge of $g^k(V_l)^{(\cap)}$ is to remove then all its predecessors on $g^k(V_l)^{(\cap)}$ are also to remove. Hence, the edges to remove on

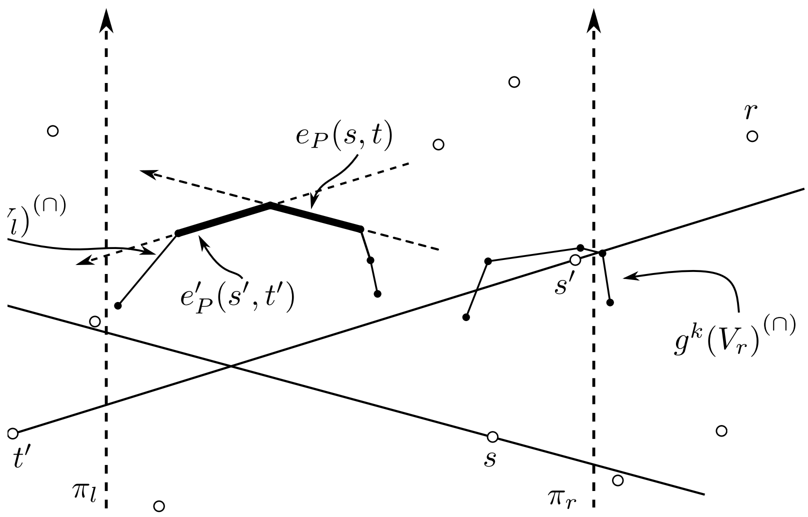

Figure 5.6: The point r is on the right of $(s't')$ and thus on the right of (st). Hence both edges $e_P(s,t)$ and $e_{P'}(s',t')$ are to remove from $g^k(V_l)^{(\cap)}$.

$g^k(V_l)^{(\cap)}$ form a connected polygonal line starting at the rightmost vertex of $g^k(V_l)$.

Similarly, the edges to remove from $g^k(V_l)^{(\cup)}$ form a connected polygonal ending in the rightmost vertex of $g^k(V_l)$. Thus, the edges to remove from $g^k(V_l)$ form a connected polygonal line containing the rightmost vertex of $g^k(V_l)$.

Using a symmetric proof, we can also find that the polygonal line to remove from $g^k(V_r)$ is also connected and contains the leftmost vertex of $g^k(V_r)$. $\square$

Denote respectively by $\mathcal{D}_l{}^{(\cap)}$ and $\mathcal{D}_r{}^{(\cap)}$ the lines to remove on $g^k(V_l)^{(\cap)}$ and $g^k(V_r)^{(\cap)}$, oriented in counter clockwise direction. That is, the rightmost vertex of $g^k(V_l)$ and the leftmost vertex of $g^k(V_r)$ are respectively the start vertex of $\mathcal{D}_l{}^{(\cap)}$ and the end vertex of $\mathcal{D}_r{}^{(\cap)}$ (see Figure 5.7).

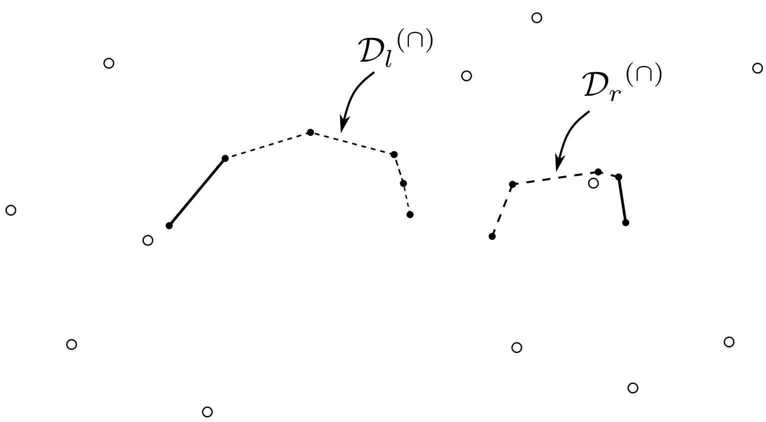

Figure 5.7: The polygonal line $\mathcal{D}_l{}^{(\cap)}$ to remove from $g^k(V_l)^{(\cap)}$ and the polygonal line $\mathcal{D}_r{}^{(\cap)}$ to remove from $g^k(V_r)^{(\cap)}$.

Now, it will be shown that the edges of ${\mathcal{D}_l}^{(\cap)}$ can be found efficiently by only traversing the edges to remove on the k-set polygons.

Property 5.12. *An edge $e_P(s,t)$ of $g^k(V_l)^{(\cap)}$ is also an edge of $g^k(V_l \cup V_r)$ if and only if the k-set T_r which is $//_{(st)}$-separable from V_r is such that $T_r \setminus V_l \subset (\overset{\circ}{st})^+$.*

Proof. Since $|T_r| = k$ and $|V_l \cap V_r| \leq k-1$, $|T_r \setminus V_l| \geq 1$. It follows, from Proposition 2.12, that if $(T_r \setminus V_l) \cap (\overset{\circ}{st})^- \neq \emptyset$, then $e_P(s,t)$ is not an edge of $g^k(V_l \cup V_r)$.

Suppose now that $T_r \setminus V_l \subset (\overset{\circ}{st})^+$. There exists a straight line Δ parallel to (st), oriented as (st), and such that $T_r \subset \mathring{\Delta}^-$ and $V_r \setminus T_r \subset \mathring{\Delta}^+$. Since $T_r \setminus V_l \subset (\overset{\circ}{st})^+$ and $|T_r \setminus V_l| \geq 1$, at least one point of T_r belongs to $(\overset{\circ}{st})^+$. Thus $\Delta \subset (\overset{\circ}{st})^+$ and $V_r \setminus T_r \subset (\overset{\circ}{st})^+$. From Proposition 2.12, it follows that $e_P(s,t)$ is an edge of $g^k(V_l \cup V_r)$ (see Figure 5.8). $\square$

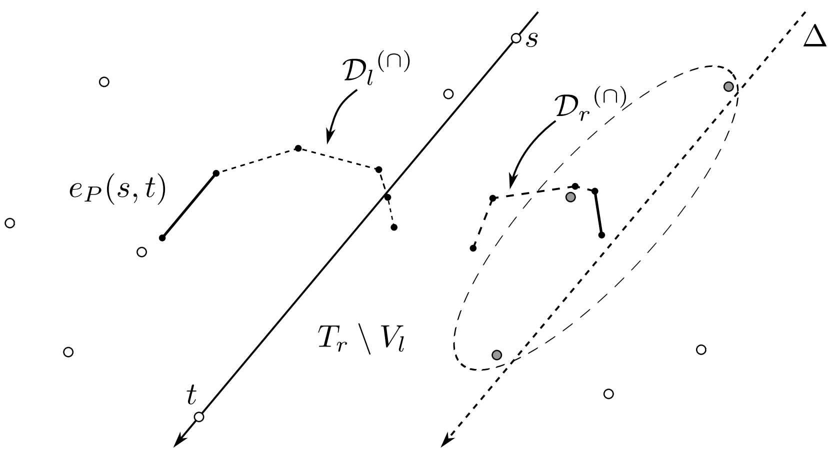

Figure 5.8: The subset $T_r \setminus V_l$ that is $//_{(st)}$-separable from V_r is on the left of (st), thus the edge $e_P(s,t)$ is an edge of $g^k(V_l \cup V_r)$.

Thus, finding the edges of ${\mathcal{D}_l}^{(\cap)}$ comes down to finding the first edge $e_P(s,t)$ of $g^k(V_l)^{(\cap)}$ that verifies Property 5.12. Now, given a straight line Δ, the k-element set T $//_\Delta$-separable from V_r can be found thanks to the following lemma:

Lemma 5.13. *If the k-set polygon $g^k(V_r)$ is not reduced to a unique vertex, let $g(T_0)$, ..., $g(T_m)$ and $e_{P_1}(s_1,t_1)$, ..., $e_{P_m}(s_m,t_m)$ be the vertices and the edges of $g^k(V_r)^{(\cap)}$, given in counter clockwise direction. Let Δ be an oriented straight line with $\theta(\Delta) \in [\pi/2, 3\pi/2]$. T_i is $//_\Delta$-separable from V_r if and only if,*
- either $i = 0$ and $\Delta <_\theta (s_1 t_1)$,
- either $i \in \{1, \ldots, m-1\}$ and $(s_i t_i) <_\theta \Delta <_\theta (s_{i+1} t_{i+1})$,
- or $i = m$ and $(s_m t_m) <_\theta \Delta$.

Proof. Since $g(T_0)$ is the rightmost vertex of $g^k(V_r)$, T_0 can be separated from V_r with a straight line Δ_0 such that $\theta(\Delta_0) = \pi/2$. The same, T_m can be separated from V_r with a straight line Δ_{m+1} such that $\theta(\Delta_{m+1}) = 3\pi/2$. For every $i \in \{1, \dots, m\}$, let $\Delta_i = (s_i t_i)$. Then, from Proposition 2.12, for every $i \in \{0, \dots, m\}$, $T_i \subset \Delta_i^- \cap \Delta_{i+1}^-$ and $V_r \setminus T_i \subset \Delta_i^+ \cap \Delta_{i+1}^+$. For every straight line Δ such that $\Delta_i <_\theta \Delta <_\theta \Delta_{i+1}$, the line Δ' parallel to Δ, oriented as Δ, and passing through $x = \Delta_i \cap \Delta_{i+1}$ is such that $\Delta_i^- \cap \Delta_{i+1}^- \subset \Delta'^-$ and $\Delta_i^+ \cap \Delta_{i+1}^+ \subset \Delta'^+$ (see Figure 5.9). It follows that T_i is $//_\Delta$-separable from V_r (if $x \in V_r$, it suffices to move slightly Δ' parallely to itself such that it strictly separates T_i from V_r).

The converse follows directly from the fact that, for a given straight line Δ with $\theta(\Delta) \in [\pi/2, 3\pi/2]$, there exists at most one set T_i of k points $//_\Delta$-separable from V_r. □

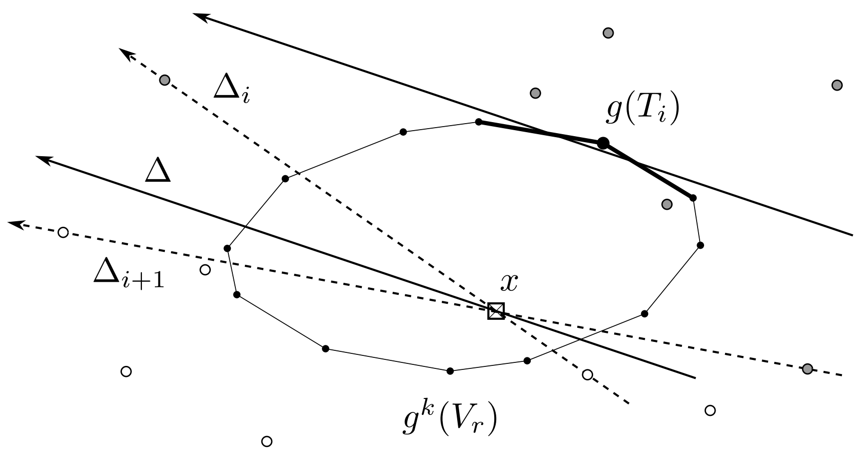

Figure 5.9: The set T_i (in gray) is Δ-separable from V_r and the vertex $g(T_i)$ is $//_\Delta$-separable from $g^k(V_r)$.

5.3.2 Edge removal algorithm

To store the k-set polygon of any subset U of V, we use a data structure similar to the one in Chapter 4 but we store independently the lines $g^k(U)^{(\cap)}$ and $g^k(U)^{(\cup)}$. For each edge $e_P(s,t)$ of $g^k(U)^{(\cap)}$, only the points s and t need to be referenced. Also, a reference to the next and previous edge on $g^k(U)^{(\cap)}$ are stored in the edge data structure.

To avoid dealing with the special cases $i = 0$ and $i = m$ of Lemma 5.13 in the algorithm, we add two anchor-edges to the upper line $g^k(U)^{(\cap)}$ in the following way (see Figure 5.10): If $g(T)$ is the rightmost vertex of $g^k(U)^{(\cap)}$, insert an anchor-edge $e_P(s,t)$ with end vertex $g(T)$, such that t is the leftmost point of T and s is any point in the plane having the same x-coordinate as t and a smaller y-coordinate (note that, from the assumptions on V, $s \notin V$). Clearly, $\theta(st) = \pi/2$. In the same way, if $g(T)$ is the leftmost vertex of $g^k(U)^{(\cap)}$, insert an anchor-edge $e_P(s,t)$ with start vertex $g(T)$, such that s is the rightmost point of T and t is any point in the plane having the same x-coordinate as s and a smaller y-coordinate (*i.e.* $\theta(st) = 3\pi/2$). Note that if $g^k(U)^{(\cap)}$ is reduced to a unique vertex, this vertex will be incident to both anchor-edges.

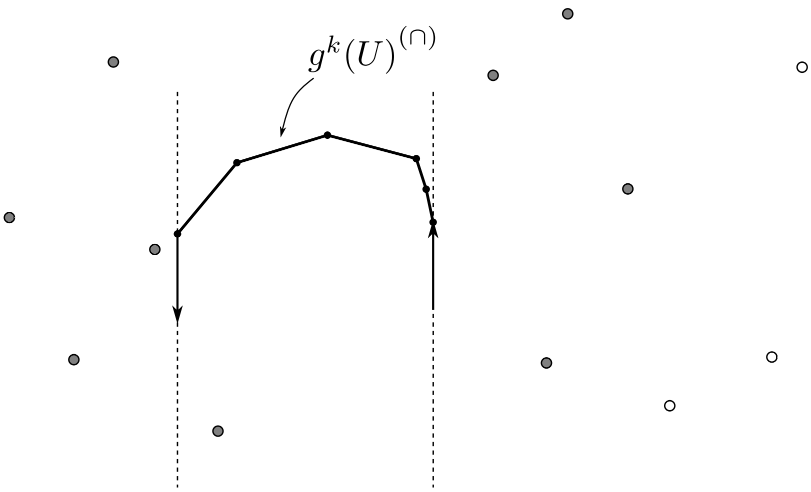

Figure 5.10: The set U (in gray) and its corresponding upper k-set polygon $g^k(U)^{(\cap)}$. The two anchor edges are inserted in the leftmost and the rightmost vertices of $g^k(U)^{(\cap)}$.

Moreover, the k-sets whose centroids are the leftmost and rightmost vertices of $g^k(U)^{(\cap)}$ are both stored in the data structure. $g^k(U)^{(\cup)}$ is represented in a symetric way.

The results of subsection 5.3.1, allow us now to give an algorithm that can find $\mathcal{D}_l{}^{(\cap)}$, by only walking the edges to remove on $g^k(V_l)^{(\cap)}$ and $g^k(V_r)^{(\cap)}$. The idea consists in starting from the leftmost vertex $g(T_{r_0})$ of $g^k(V_r)^{(\cap)}$ and from the edge out of the rightmost vertex of $g^k(V_l)^{(\cap)}$. First, we try to find the first edge $e_{P_{l_i}}(s_{l_i}, t_{l_i})$ of $g^k(V_l)^{(\cap)}$ that is not invalidated by $g(T_{r_0})$, that is, such that $T_{r_0} \setminus V_l \subset (s_{l_i}\mathring{} t_{l_i})^+$. From Lemma 5.10, no edge of $g^k(V_l)^{(\cap)}$ that succeeds $e_{P_{l_i}}(s_{l_i}, t_{l_i})$ is invalidated by $g(T_{r_0})$. Thus, the points of T_{r_0} do not have to be considered afterward by the algorithm and we can pass to the predecessor $g(T_{r_1})$ of $g(T_{r_0})$ on $g^k(V_r)^{(\cap)}$. As in the previous step, we seek now the first edge of $g^k(V_l)^{(\cap)}$ that succeeds $e_{P_{l_i}}(s_{l_i}, t_{l_i})$ and that is not to be invalidated by $g(T_{r_1})$. $g(T_{r_1})$ invalidate none of the next edges on $g^k(V_l)^{(\cap)}$ and we can pass to the predecessor of $g(T_{r_1})$ on $g^k(V_r)^{(\cap)}$. And so on.

Suppose now that the algorithm has just found the first edge $e_{P_{l_j}}(s_{l_j}, t_{l_j})$ not invalidated by a given vertex $g(T_{r_h})$ of $g^k(V_r)^{(\cap)}$ and that the edge $e_{P_{r_h}}(s_{r_h}, t_{r_h})$

entering in $g(T_{r_h})$ is such that $s_{l_j}t_{l_j} <_\theta s_{r_h}t_{r_h}$. In this case, the algorithm can stop because $e_{P_{l_j}}(s_{l_j}, t_{l_j})$ is the first edge that should not be removed from $g^k(V_l)^{(\cap)}$. Indeed, by construction, all the edges preceding $e_{P_{l_j}}(s_{l_j}, t_{l_j})$ on $g^k(V_l)^{(\cap)}$ are invalidated by at least one vertex of $g^k(V_r)^{(\cap)}$. Moreover, from Lemma 5.13, if $s_{l_j}t_{l_j} <_\theta s_{r_h}t_{r_h}$, the vertex of $g^k(V_r)^{(\cap)}$ that is $//_{(s_{l_j}t_{l_j})}$-separable from $g^k(V_r)$ is one of the vertices $g(T_{r_f})$ that was already checked by the algorithm. Then $e_{P_{l_j}}(s_{l_j}, t_{l_j})$ is not invalidated by $g(T_{r_f})$, that is $T_{r_f} \setminus V_l \subset (s_{l_j}t_{l_j})^+$. From Property 5.12, $e_{P_{l_j}}(s_{l_j}, t_{l_j})$ is not to be removed.

Note that the algorithm will necessarily stop, at the latest when $e_{P_{l_j}}(s_{l_j}, t_{l_j})$ is the anchor edge incident in the leftmost vertex of $g^k(V_l)^{(\cap)}$ or when $e_{P_{r_h}}(s_{r_h}, t_{r_h})$ is the anchor edge incident in the rightmost vertex of $g^k(V_r)^{(\cap)}$. In the first case, the angle $\theta(s_{l_j}t_{l_j}) = 3\pi/2$ is greater than the angles θ of all the edges of $g^k(V_r)^{(\cap)}$ and all the points of $V_r \setminus V_l$ are on the left of $(s_{l_j}t_{l_j})$. In the second case, the angle $\theta(s_{r_h}t_{r_h}) = \pi/2$ is smaller than the angles θ of all the edges of $g^k(V_l)^{(\cap)}$. The general form of the algorithm that obtains $\mathcal{D}_l$ is then the following:

```
function find_D_l^(∩)
{
  let e_P(s,t) be the edge of g^k(V_l)^(∩) with start vertex the
      rightmost vertex of g^k(V_l)^(∩);
  let g(T) be the leftmost vertex of g^k(V_r);
  let e_P'(s',t') be the edge of g^k(V_r)^(∩) with end vertex g(T) (i.e.
      P' ∪ {t'} = T);
  1. while (T \ V_l ⊄ (st)^+)
       e_P(s,t) ⟵ successor of e_P(s,t) on g^k(V_l);
  2. while (st <_θ s't')
  {
       e_P'(s',t') ⟵ predecessor of e_P'(s',t') on g^k(V_r);
       3. while (P' ∪ {t'}) \ V_l ⊄ (st)^+
            e_P(s,t) ⟵ successor of e_P(s,t) on g^k(V_l);
  }
  return e_P(s,t);
}
```

Proposition 5.14. *The function* `find_`$D_l^{(\cap)}$ *can be implemented to run in time* $O(k + |\mathcal{D}_l^{(\cap)}| \log k + |\mathcal{D}_r^{(\cap)}|)$.

Proof. (i) As explained above, from Lemmas 5.10 and 5.13 and from Property 5.12, within a margin of one, only the edges to remove are traversed on $g^k(V_l)^{(\cap)}$. Moreover, for every edge $e_{P'}(s', t')$ of $g^k(V_r)^{(\cap)}$ traversed by the algorithm, except for the last one, there exists an edge $e_P(s,t)$ of $g^k(V_l)^{(\cap)}$ such that $st <_\theta s't'$ (loop (2) condition) and $(P' \cup \{t'\}) \setminus V_l \subset (\mathring{st})^+$ (loops (1) and (3) conditions). Since, $(P' \cup \{t'\}) \setminus V_l$ contains at least one point and since this point belongs to $\mathring{\pi}_r^- \cap (\mathring{st})^+$, $(st) \cap (s't') \in \mathring{\pi}_r^-$. Thus $(st)^- \cap \mathring{\pi}_l^+ \subset (\mathring{s't'})^-$ and, since $(st)^- \cap \mathring{\pi}_l^+$ contains at least one point of $(P \cup \{s,t\}) \setminus V_r$, $e_{P'}(s',t')$ is not an edge of $g^k(V_l \cup V_r)$. It follows that, within a margin of one, only the edges to remove are traversed on $g^k(V_r)^{(\cap)}$.

(ii) In loop 1, $g(T)$ is the leftmost vertex of $g^k(V_r)$ and, by hypothesis, T is stored in the data structure containing $g^k(V_r)$. To check whether $T \setminus V_l \subset (\mathring{st})^+$, it suffices to compute the straight line passing through s, tangent to the convex hull $conv(T \setminus V_l)$ at a point r, and such that $conv(T \setminus V_l) \subset (rs)^+$ (see Figure 5.11). $(T \setminus V_l) \subset (\mathring{st})^+$ is then equivalent to $r \in (\mathring{st})^+$. Since the points of V are sorted from left to right, $T \setminus V_l$ can be obtained in $O(k)$ time and $conv(T \setminus V_l)$ can also be computed in $O(k)$ time. Any tangent to $conv(T \setminus V_l)$ can then be found in $O(\log k)$ time (see for example [OvL81]). The time complexity of loop 1 of the algorithm can thus be bounded by $O(k + |\mathcal{D}_l{}^{(\cap)}| \log k)$.

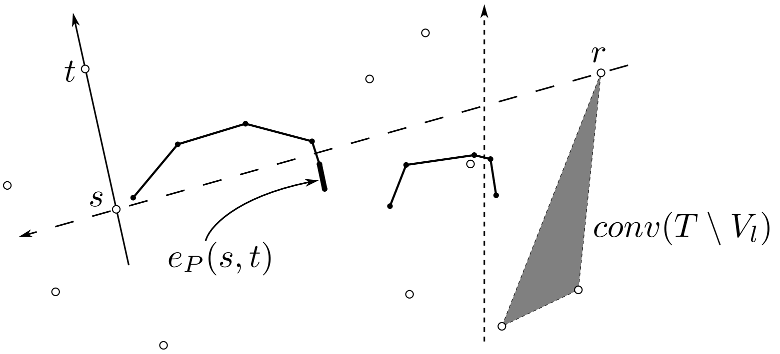

Figure 5.11: Illustration of the proof of Proposition 5.14 (ii).

(iii) From (i), the test $(P' \cup \{t'\}) \setminus V_l \not\subset (\mathring{st})^+$ in loop 3 condition is done at most $|\mathcal{D}_l{}^{(\cap)}| + |\mathcal{D}_r{}^{(\cap)}| + 2$ times. From Lemma 5.10, given a set $P' \cup \{t'\}$, if an edge $e_P(s,t)$ of $g^k(V_l)^{(\cap)}$ is such that $(P' \cup \{t'\}) \setminus V_l \subset (\mathring{st})^+$, then all its successors verify the same inclusion. Furthermore, if $e_{P''}(s'',t'')$ is the predecessor of an edge $e_{P'}(s',t')$ of $g^k(V_r)$, then $P'' \cup \{t''\} = P' \cup \{s'\}$, from Proposition 2.12. It follows that, for two consecutive passes in loop 2, the considered sets $(P' \cup \{t'\}) \setminus V_l$ differ from each other by at most one point and the test $(P' \cup \{t'\}) \setminus V_l \not\subset (\mathring{st})^+$ can be achieved in constant time. It is the same with all the other instructions of loop 2, which are all together in $O(|\mathcal{D}_l{}^{(\cap)}| + |\mathcal{D}_r{}^{(\cap)}|)$. □

Obviously, $\mathcal{D}_r{}^{(\cap)}$ can be found in a symmetric way and it is the same with the lines to remove on $g^k(V_l)^{(\cup)}$ and $g^k(V_r)^{(\cup)}$. Hence the theorem:

Theorem 5.15. *The edges to remove on $g^k(V_l)$ and $g^k(V_r)$ can be found in time $O(k + d \log k)$, where d is the total number of edges to remove.*

5.3.3 Edge construction

In the preceding section, it has been shown that the edges to remove form two connected lines on the left and on the right k-set polygons. Since the k-set polygon to construct is convex, the edges to create form also two connected lines, an upper and a lower one. $\mathcal{C}^{(\cap)}$ denotes the oriented upper line to create and $\mathcal{C}^{(\cup)}$ denotes the lower line (see Figure 5.12). $\mathcal{C}^{(\cap)}$ connects the start vertex of ${\mathcal{D}_r}^{(\cap)}$ to the end vertex of ${\mathcal{D}_l}^{(\cap)}$ (obtained by Algorithm `find_`${\mathcal{D}_l}^{(\cap)}$).

We show now that the vertices of $\mathcal{C}^{(\cap)}$ can be found by considering k-set polygons of at most $2k$ points of $V_l \cup V_r$.

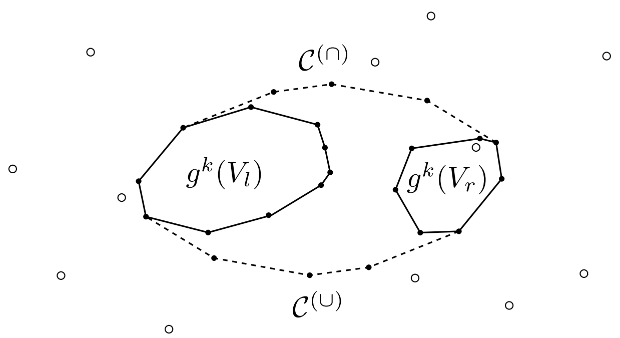

Figure 5.12: Uppon joining $g^k(V_l)$ and $g^k(V_r)$ two polygonal lines are created: $\mathcal{C}^{(\cap)}$ and $\mathcal{C}^{(\cup)}$.

Property 5.16. *(i) Let $g(T)$ be a vertex of $\mathcal{C}^{(\cap)}$ and let Δ be an oriented straight line that is not parallel to any other straight line passing through two points of $V_l \cup V_r$ and such that T is $//_\Delta$-separable from $V_l \cup V_r$.*

The subsets T_l and T_r of k points that are $//_\Delta$-separable from V_l and V_r respectively are such that $g(T_l)$ is a vertex of ${\mathcal{D}_l}^{(\cap)}$, $g(T_r)$ is a vertex of ${\mathcal{D}_r}^{(\cap)}$, and $g(T)$ is the vertex of ${g^k(T_l \cup T_r)}^{(\cap)}$ that is $//_\Delta$-separable from $g^k(T_l \cup T_r)$.

(ii) Conversely, let $g(T_l)$ be a vertex of ${\mathcal{D}_l}^{(\cap)}$ and $g(T_r)$ be a vertex of ${\mathcal{D}_r}^{(\cap)}$ such that there exists a straight line Δ that is not parallel to any straight line passing through two points of $V_l \cup V_r$ and such that T_l and T_r are $//_\Delta$-separable from V_l and from V_r respectively. If T is the k-point subset that is $//_\Delta$-separable from $T_l \cup T_r$ then $g(T)$ is a vertex of $\mathcal{C}^{(\cap)}$ and is $//_\Delta$-separable from $g^k(V_l \cup V_r)$.

Proof. (i) Let Δ' and Δ_l be two straight lines parallel to Δ and such that T and T_l are respectively Δ'-separable from $V_l \cup V_r$ and Δ_l-separable from V_l. Suppose that there exists $v \in T \setminus (T_l \cup T_r)$. Within a permutation of V_r and V_l, we can assume that $v \in V_l$. Since $v \notin T_l$, $\Delta_l \subset \mathring{\Delta}'^-$ and, therefore, $T_l \cup \{v\} \subset \mathring{\Delta}'^-$ and $T_l \cup \{v\} \subseteq T$ which is absurd since $|T| = |T_l| = k$. It follows that $T \subseteq T_l \cup T_r$. T is then a set Δ'-separable from $T_l \cup T_r$ and, from Proposition 2.11, $g(T)$ is a vertex of $g^k(T_l \cup T_r)$ $//_\Delta$-separable from $g^k(T_l \cup T_r)$. It follows that $g(T)$ is a vertex of $g^k(T_l \cup T_r)^{(\cap)}$.

Moreover, if $g(T_r)$ is the leftmost vertex of $g^k(V_r)$, from Property 5.9, it belongs to $\mathcal{D}_r{}^{(\cap)}$. Otherwise, from Lemma 5.13, the edge $e_P(s,t)$ of $g^k(V_r)$ with start vertex $g(T_r)$ is such that $\Delta <_\theta (st)$. Then $e_P(s,t)$ cannot be an edge of $g^k(V_l \cup V_r)$ since it should precede $g(T)$ on $g^k(V_l \cup V_r)^{(\cap)}$. It follows that $g(T_r)$ is a vertex of $\mathcal{D}_r{}^{(\cap)}$. In the same way, $g(T_l)$ is a vertex of $\mathcal{D}_l{}^{(\cap)}$.

(ii) Conversely, let Δ' be the straight line parallel to Δ, with the same orientation as Δ, that passes through a point of T, and such that $T \subset \Delta'^-$. By construction, $(V_l \cup V_r) \setminus T$ is then in $\mathring{\Delta}'^+$. Let Δ_l be an oriented straight line parallel to Δ and such that T_l is Δ_l-separable from V_l.

Since $|T_l| = |T|$, at least one point of T_l belongs to Δ'^+ and it results that $\Delta_l \subset \mathring{\Delta}'^+$. Hence $V_l \setminus T_l \subset \mathring{\Delta}'^+$. In the same way $V_r \setminus T_r \subset \mathring{\Delta}'^+$. Since, by construction, $T_l \setminus T$ and $T_r \setminus T$ are also in $\mathring{\Delta}'^+$, it results that $g(T)$ is a vertex of $g^k(V_l \cup V_r)$ that is $//_\Delta$-separable from $g^k(V_l \cup V_r)$. It follows that $g(T)$ is a vertex of $g^k(V_l \cup V_r)^{(\cap)}$.

Moreover, since $g(T_r)$ is a vertex of $\mathcal{D}_r{}^{(\cap)}$, every edge $e_{P_l}(s_l,t_l)$ of $g^k(V_l)^{(\cap)}$ that is also an edge of $g^k(V_l \cup V_r)$ is such that $(s_l t_l) <_\theta \Delta$. In the same way, for every edge $e_{P_r}(s_r,t_r)$ of $g^k(V_r)^{(\cap)}$ that is also an edge of $g^k(V_l \cup V_r)$, $\Delta <_\theta (s_r t_r)$. It results that $g(T)$ is a vertex of $\mathcal{C}^{(\cap)}$ (see Figure 5.13). □

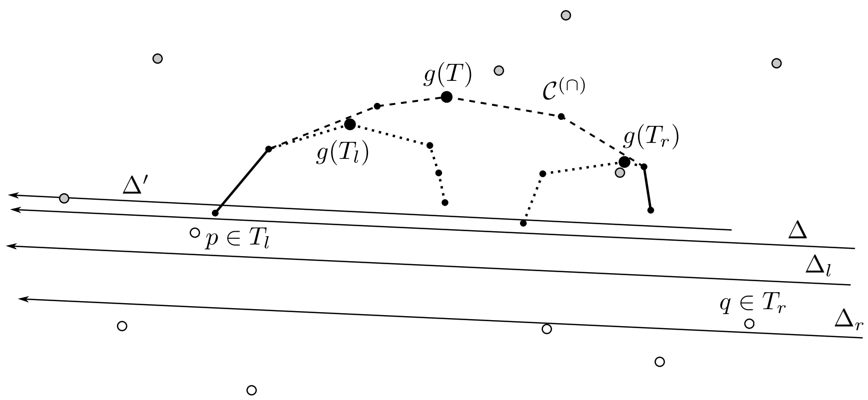

Figure 5.13: The set T that is Δ-separable from $V = V_l \cup V_r$ and the vertices $g(T_l)$ of $\mathcal{D}_l{}^{(\cap)}$ and $g(T_r)$ of $\mathcal{D}_r{}^{(\cap)}$ that are $//_\Delta$-separable from $g^k(V_l)$ and $g^k(V_r)$ respectively.

It follows from this proposition that, to construct $\mathcal{C}^{(\cap)}$, we have to consider all the couples of vertices $(g(T_l), g(T_r))$, where $g(T_l)$ and $g(T_r)$ belong to $\mathcal{D}_l{}^{(\cap)}$ and

${\mathcal{D}_r}^{(\cap)}$ respectively and such that T_l and T_r are $//_\Delta$-separable from V_l and V_r with a same straight line Δ. Then it suffices, for each of these couples, to compute the k-set polygon of $T_l \cup T_r$ and to extract some of its vertices. These couples can be generated efficiently by using the result of Lemma 5.13.

Indeed, let $g(T_{min})$ be the first vertex of ${\mathcal{D}_r}^{(\cap)}$ and $g(T_{max})$ be the last vertex of ${\mathcal{D}_l}^{(\cap)}$. $g(T_{min})$ and $g(T_{max})$ are then also the first and the last endpoints of $\mathcal{C}^{(\cap)}$ (see Figure 5.14).

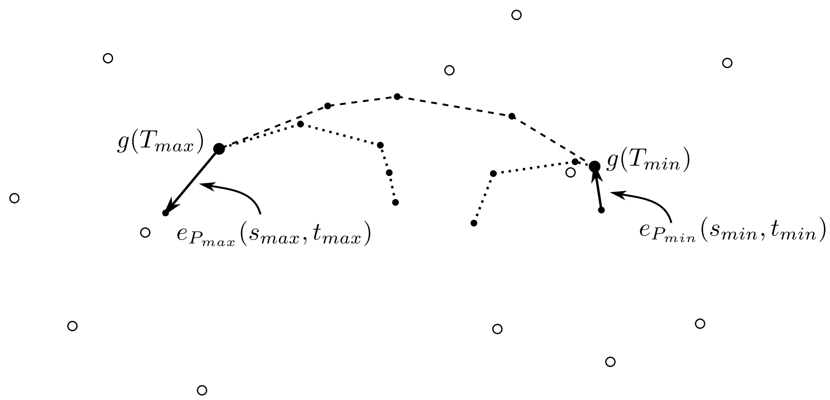

Figure 5.14: The vertices $g(T_{min})$ and $g(T_{max})$; also the edges $e_{P_{min}}(s_{min}, t_{min})$ entering $g(T_{min})$ and $e_{P_{max}}(s_{max}, t_{max})$ leaving $g(T_{max})$.

From Lemma 5.13, if $e_{P_{min}}(s_{min}, t_{min})$ and $e_{P_{max}}(s_{max}, t_{max})$ are the edges (may be anchor edges) respectively entering $g(T_{min})$ on $g^k(V_r)^{(\cap)}$ and leaving $g(T_{max})$ on $g^k(V_l)^{(\cap)}$, then the vertices of $\mathcal{C}^{(\cap)}$ are the vertices $//_\Delta$-separable from $g^k(V_l \cup V_r)$, with $(s_{min}t_{min}) <_\theta \Delta <_\theta (s_{max}t_{max})$. To find all the vertices of $\mathcal{C}^{(\cap)}$, we need then to generate all the couples of vertices $(g(T_l), g(T_r))$ such that T_l and T_r are $//_\Delta$-separable from V_l and V_r respectively, with $(s_{min}t_{min}) <_\theta \Delta <_\theta (s_{max}t_{max})$.

For every vertex $g(T_l)$ of ${\mathcal{D}_l}^{(\cap)}$ let $e_{P_l}(s_l, t_l)$ and $e_{P'_l}(s'_l, t'_l)$ be the edges of $g^k(V_l)^{(\cap)}$ entering and leaving $g(T_l)$. For every vertex $g(T_r)$ of ${\mathcal{D}_r}^{(\cap)}$ let $e_{P_r}(s_r, t_r)$ and $e_{P'_r}(s'_r, t'_r)$ be the edges of $g^k(V_r)^{(\cap)}$ entering and leaving $g(T_r)$.

From Lemma 5.13, T_l and T_r are then $//_\Delta$-separable from V_l and V_r (with the same straight line Δ), if and only if, the intervals $I_l = [\theta(s_l t_l), \theta(s'_l t'_l)]$ associated with T_l (or $g(T_l)$) and $I_r = [\theta(s_r t_r), \theta(s'_r t'_r)]$ associated with T_r (or $g(T_r)$) have a non empty intersection (note that in the case of the first vertices and of the last vertices of ${\mathcal{D}_l}^{(\cap)}$ and of ${\mathcal{D}_r}^{(\cap)}$, the anchor edges may be used to form the intervals). We note by $\theta(g(T_l)g(T_r))$ this intersection *i.e.*,

$$\theta(g(T_l)g(T_r)) = [max(\theta(s_l t_l), \theta(s_r t_r)), min(\theta(s'_l t'_l), \theta(s'_r t'_r))].$$

Let $(I_{l_1}, \ldots, I_{l_m})$ be the sequence of intervals associated to the vertices $g(T_l)$ of ${\mathcal{D}_l}^{(\cap)}$ in the ascending order and such that the vertices $g(T_l)$ are $//_\Delta$-separable from ${\mathcal{D}_l}^{(\cap)}$ by straight lines Δ such that $s_{min}t_{min} <_\theta \Delta <_\theta s_{max}t_{max}$. Similarly, let $(I_{r_1}, \ldots, I_{r_{m'}})$ be the sequence of intervals associated to the vertices $g(T_r)$ of ${\mathcal{D}_r}^{(\cap)}$ in the ascending order and such that the vertices $g(T_r)$ are $//_\Delta$-separable from ${\mathcal{D}_l}^{(\cap)}$ by the same straight lines Δ. Obviously both $(I_{l_1}, \ldots, I_{l_m})$ and $(I_{r_1}, \ldots, I_{r_{m'}})$ are monotonous and the non empty intersections of the sets $I_{l_1} \cap I_{r_1}, I_{l_1} \cap I_{r_2}, \ldots, I_{l_2} \cap I_{r_1}, I_{l_2} \cap I_{r_2}, \ldots, I_{l_m} \cap I_{r_{m'-1}}, I_{l_m} \cap I_{r_{m'}}$ form also a monotonous ascending partition of $[\theta(s_{min}t_{min}), \theta(s_{max}t_{max})]$. Let $(\theta_1, \ldots, \theta_{m''})$ be the sequence of these intersections in the ascending order. Note that $\theta_1 = \theta(g(T_l)g(T_{min}))$ with $g(T_l)$ the vertex of ${\mathcal{D}_l}^{(\cap)}$ that is $//_{(s_{min}t_{min})}$-separable from ${\mathcal{D}_l}^{(\cap)}$ and that $\theta_{m''} = \theta(g(T_{max})g(T_r))$ with $g(T_r)$ the vertex of ${\mathcal{D}_r}^{(\cap)}$ that is $//_{(s_{max}t_{max})}$-separable from ${\mathcal{D}_r}^{(\cap)}$.

We start constructing the polygonal line $\mathcal{C}^{(\cap)}$ in the clockwise direction. For that, we need to obtain the intervals $(\theta_1, \ldots, \theta_{m''})$ in this order. The first vertex to appear on $\mathcal{C}^{(\cap)}$ is the vertex $g(T_{min})$. Since $g(T_{min})$ is a vertex of ${\mathcal{D}_r}^{(\cap)}$ we need to find the first vertex $g(T_l)$ on ${\mathcal{D}_l}^{(\cap)}$ that is $//_\Delta$-separable from $g^k(V_l)$ by a straight line Δ that separates $g(T_{min})$ from $g^k(V_r)$. We start looking for the vertex $g(T_l)$ starting from the rightmost vertex of ${g^k(V_l)}^{(\cap)}$ and advancing till we reach a vertex $g(T_l)$ such that $\theta(g(T_l)g(T_{min})) \neq \emptyset$. The interval $\theta(g(T_l)g(T_{min}))$ defines the first interval θ_1. Using Property 5.16, all the vertices $g(T)$ of ${g^k(T_l \cup T_{min})}^{(\cap)}$ that are $//_\Delta$-separable from ${g^k(T_l \cup T_{min})}^{(\cap)}$ by some straight line Δ with $\theta(\Delta) \in \theta_1$ are vertices of $\mathcal{C}^{(\cap)}$. It suffices then to take these vertices as they appear on ${g^k(T_l \cup T_{min})}^{(\cap)}$ and add them to $\mathcal{C}^{(\cap)}$ starting from $g(T_{min})$. We say that the interval θ_1 generates these vertices.

Now we need to find the next interval θ_2, or to put it in another way, the next couple $(g(T_l), g(T_r))$. If the edges $s'_l t'_l$ and $s'_r t'_r$ leaving $g(T_l)$ and $g(T_r)$ respectively are such that $s'_l t'_l <_\theta s'_r t'_r$ then we keep for $g(T_r)$ the vertex $g(T_{min})$ and we move to the next vertex $g(T_l)$ on ${\mathcal{D}_l}^{(\cap)}$, otherwise we keep $g(T_l)$ and we move to the next vertex $g(T_r)$ on ${\mathcal{D}_r}^{(\cap)}$. Similarly to what we have done earlier, all the vertices of ${g^k(T_l \cup T_r)}^{(\cap)}$ that are $//_\Delta$-separable from $g^k(T_l \cup T_r)$ with $\theta(\Delta) \in \theta_2$ are vertices of $\mathcal{C}^{(\cap)}$, from Property 5.16. We walk in this way on ${\mathcal{D}_l}^{(\cap)}$ and ${\mathcal{D}_r}^{(\cap)}$ till we reach the last interval $\theta_{m''}$ with the last couple $(g(T_{max}), g(T_r))$. This interval will generate the last part of $\mathcal{C}^{(\cap)}$.

We show now how to find efficiently the first vertex generated by every interval θ_i:

Lemma 5.17. *(i) $g(T_{min})$ is the first vertex generated by the interval θ_1.*
(ii) Given two successive intervals θ_i and θ_{i+1}, the last vertex generated by θ_i is equal to the first vertex generated by θ_{i+1}.

Proof. (i) From Lemma 5.13, since $e_{P_{min}}(s_{min}, t_{min})$ is an edge of ${g^k(V_l \cup V_r)}^{(\cap)}$, $T_{min} = P_{min} \cup \{t_{min}\}$ is $//_\Delta$-separable from $V_l \cup V_r$ by a straight line Δ that tends toward $(s_{min}t_{min})$ and such that $(s_{min}t_{min}) <_\theta \Delta$. Since $\theta(s_{min}t_{min})$ is the lower bound of θ_1, it follows that $\theta(\Delta) \in \theta_1$ and that $g(T_{min})$ is the first vertex generated by θ_1.

(ii) By construction, there exists an edge $e_P(s,t)$ of $g^k(V_l)$ or $g^k(V_r)$ such that $\theta(st)$ is the limit value between θ_i and θ_{i+1}. If $g(T)$ is the last vertex generated by θ_i, T is $//_\Delta$-separable from $V_l \cup V_r$ with a straight line Δ that tends toward (st) and such that $\Delta <_\theta (st)$. In the same way, for the first vertex $g(T')$ generated by θ_{i+1}, T' is $//_{\Delta'}$-separable from $V_l \cup V_r$ with a straight line Δ' that tends toward (st) and such that $(st) <_\theta \Delta'$. Since no four points of V are supposed to belong to tow parallel lines, it follows that T and T' are both $//_{(st)}$-separable from $V_l \cup V_r$ and, hence, $g(T) = g(T')$. □

It results from this lemma that, if (T_l, T_r) and (T'_l, T'_r) are two consecutively treated couples, then the last vertex extracted from $g^k(T_l \cup T_r)$ is equal to the first vertex to extract from $g^k(T'_l \cup T'_r)$. Thus it suffices to maintain a link to this vertex in the algorithm while constructing $g^k(T'_l \cup T'_r)$ from $g^k(T_l \cup T_r)$.

Hence, we can give the whole algorithm that constructs $\mathcal{C}^{(\cap)}$:

```
function construct_C^(∩)
{
  let e_{P_min}(s_min, t_min) be the edge of g^k(V_r)^(∩) ending at the start
      vertex of D_r^(∩);
  let e_{P_max}(s_max, t_max) be the edge of g^k(V_l)^(∩) starting at the end
      vertex of D_l^(∩);
  let e_{P_l}(s_l, t_l) be the edge of g^k(V_l)^(∩) starting at the start
      vertex of D_l^(∩);
  1. while(s_l t_l <_θ s_min t_min)
       e_{P_l}(s_l, t_l) ⟵ successor of e_{P_l}(s_l, t_l) on g^k(V_l)^(∩);

  T_l ⟵ P_l ∪ {s_l};
  e_{P_r}(s_r, t_r) ⟵ successor of e_{P_min}(s_min, t_min) on g^k(V_r)^(∩);
  T_r ⟵ P_r ∪ {s_r};
  T ⟵ T_r;

  2. do
  {
       let Θ be the interval θ(g(T_l), g(T_r));

       3. let e_P(s,t) be the edge of g^k(T_l ∪ T_r) starting at g(T);
```

```
        4. while (θ(st) ∈ Θ)
        {
             insert e_P(s,t) in g^k(V_l ∪ V_r)^(∩) such that it starts at
                 g(T);
             5. T ⟵ P ∪ {t};
             let e_P(s,t) be the edge of g^k(T_l ∪ T_r) starting at g(T);
        }
        6. if (s_l t_l <_θ s_r t_r)
        {
             e_{P_l}(s_l,t_l) ⟵ successor of e_{P_l}(s_l,t_l) on g^k(V_l)^(∩);
             T_l ⟵ P_l ∪ {t_l};
        } else
        {
             e_{P_r}(s_r,t_r) ⟵ successor of e_{P_r}(s_r,t_r) on g^k(V_r)^(∩);
             T_r ⟵ P_r ∪ {s_r};
        }
    } while (θ(s_max t_max) ∉ Θ);
}
```

Proposition 5.18. *The algorithm which constructs $\mathcal{C}^{(\cap)}$ can be implemented to run in $O((k + |\mathcal{D}_r{}^{(\cap)}| + |\mathcal{D}_l{}^{(\cap)}| + |\mathcal{C}^{(\cap)}|)\log^2 k)$ time.*

Proof. The essential step of the algorithm, given a vertex $g(T)$ of $g^k(T_l \cup T_r)$, is to determine the edge $e_P(s,t)$ of $g^k(T_l \cup T_r)$ starting at $g(T)$. If the convex hulls of T and $(T_l \cup T_r) \setminus T$ are given, it suffices to find the common oriented tangent Δ of these convex hulls such that $T \subset \Delta^-$, $(T_l \cup T_r) \setminus T \subset \Delta^+$, and $s = T \cap \Delta$ precedes $t = ((T_l \cup T_r) \setminus T) \cap \Delta$ on Δ. Indeed, from Proposition 2.12, $e_{T\setminus\{s\}}(s,t)$ is then the edge of $g^k(T_l \cup T_r)$ starting at $g(T)$. We have thus to maintain the convex hulls of T and $(T_l \cup T_r) \setminus T$ all along the algorithm.

At the beginning of the algorithm, $g(T) = g(T_r)$ is the start vertex of $\mathcal{D}_r{}^{(\cap)}$. The convex hull of T can then be obtained in the following way: If $g(T')$ is the leftmost vertex of $g^k(T_r)$, the convex hull of T' can be directly computed since the points of T' are stored in the data structure containing $g^k(T_r)$. Let CH be this convex hull. $\mathcal{D}_r{}^{(\cap)}$ can then be traversed from $g(T')$ to $g(T)$ and, for each traversed edge $e_P(s,t)$, t is removed from CH and s is inserted in CH. When arriving in $g(T)$, CH contains the convex hull of T. Using the fully dynamic convex hull data structure of Overmars and van Leeuwen [OvL81], the convex hull of T' can be stored in CH in $O(k \log^2 k)$ time and every insertion or deletion in CH can be done in $O(\log^2 k)$ time. The convex hull of the set T at the beginning of the algorithm can thus be obtained in $O((k + |\mathcal{D}_r{}^{(\cap)}|)\log^2 k)$ time. Since, at the beginning of the algorithm, $T = T_r$, the convex hull of $(T_l \cup T_r) \setminus T = T_l \setminus T$ can be computed in the same way (in a dynamic structure CH') while traversing ${\mathcal{D}_l}^{(\cap)}$ in loop 1. In order to place in CH' only the points of T_l that are not in T, it suffices to mark the points of T (for example, while constructing their convex hull). During the execution of the algorithm, the set T is only modified by instruction 5. To update CH, we have just to remove s and to insert t. Since instruction 5 happens exactly once per edge created on $\mathcal{C}^{(\cap)}$, the overall complexity of all the updates of CH is $O(|\mathcal{C}^{(\cap)}| \log^2 k)$. The set $(T_l \cup T_r) \setminus T$ is modified by instructions 5 and 6. In the same way, for each of these instructions, at most one point is removed from CH' and at most one point is inserted (a point that already belongs to CH is neither removed nor inserted in CH'). Since the total number of passes in loop 2 is at most $|{\mathcal{D}_r}^{(\cap)}| + |{\mathcal{D}_l}^{(\cap)}|$ and since the total number of passes in loop 4 is equal to the number of edges of $\mathcal{C}^{(\cap)}$, it follows that the overall complexity of the updates of CH' is $O((|{\mathcal{D}_r}^{(\cap)}| + |{\mathcal{D}_l}^{(\cap)}| + |\mathcal{C}^{(\cap)}|) \log^2 k)$. Since a common tangent of T and $(T_l \cup T_r) \setminus T$ can also be found in $O(\log^2 k)$ time using CH and CH', $\mathcal{C}^{(\cap)}$ can be constructed in $O((k + |{\mathcal{D}_r}^{(\cap)}| + |{\mathcal{D}_l}^{(\cap)}| + |\mathcal{C}^{(\cap)}|) \log^2 k)$ time. □

Obviously, the lower polygonal line can be constructed similarly and we get the following result:

Theorem 5.19. *The edges to construct while merging $g^k(V_l)$ and $g^k(V_r)$ can be found in $O((k + d + c) \log^2 k)$ time, where d and c are the numbers of edges to delete and to create.*

The divide and conquer construction of the k-set polygon of a set V of at least k points is then as follows:

```
function divide_and_conquer_ksp
{
  if(|V| ≤ k+1)
  {
      construct directly g^k(V);
  }
  else
  {
      if(|V| < 2(k+1))
      {
        divide V into two non-disjoint subsets V_l and V_r of k o
            r k+1 points each such that V_l ∩ V_r belong to a
            vertical strip separating V_l \ V_r and V_r \ V_l;
      }
      else
```

```
      {
        divide V into two disjoint subsets V_l and V_r of ⌈|V|/2⌉
          and ⌊|V|/2⌋ points separable by a vertical straight
          line;
      }
      construct recursively g^k(V_l) and g^k(V_r);
      merge g^k(V_l) and g^k(V_r) with the previous algorithms;
  }
}
```

Theorem 5.20. *Algorithm* `divide_and_conquer_ksp` *constructs the k-set polygon of n points in $O(n\log n + m\log^2 k\log(n/k))$ time, where m is the worst case size of the output.*

Proof. If $n \leq k+1$, the algorithm directly constructs the k-set polygon of V. If $n = k$, this k-set polygon is reduced to the centroid $g(V)$. If $n = k+1$, from Proposition 2.11, a vertex of the k-set polygon of V is the centroid of a subset of k points of V separable from the last one by a straight line. This last point is then a vertex of the convex hull of V and it follows that constructing the k-set polygon of V comes to constructing its convex hull. This can be done in $O(k)$ time since V is sorted.

If $n > k+1$, V is divided into two subsets V_l and V_r such that $|V_l| = \lceil n/2\rceil$ and $|V_r| = \lfloor n/2\rfloor$ if $n \geq 2(k+1)$, and $|V_l| \leq k+1$ and $|V_r| \leq k+1$ otherwise. The k-set polygons of V_l and V_r are then recursively constructed and, finally, merged in $O((k+d+c)\log^2 k)$ time, where d and c are the total numbers of edges deleted and constructed in the merging step (Theorem 5.19).

Now, Dey [Dey98] and Tóth [Tót01] have shown that the size of a k-set polygon of n points is in $O(n\beta(k))$, with $2^{\Omega(\sqrt{\log k})} \leq \beta(k) \leq O(k^{1/3})$. It follows that d and c are bounded by $O(n\beta(k))$ and that the complexity of the merging is $O(n\beta(k)\log^2 k)$. Hence the induction relation that gives the complexity $T(n)$ of the algorithm (without the sorting step):

$$
\begin{array}{ll}
T(n) \leq T(\lceil n/2\rceil) + T(\lfloor n/2\rfloor) + O(n\beta(k)\log^2 k) & \text{if } n \geq 2(k+1) \\
T(n) \leq 2T(k+1) + O(n\beta(k)\log^2 k) & \text{if } k+1 < n < 2(k+1) \\
T(n) = O(k) & \text{if } n \leq k+1
\end{array}
$$

Solving this relation, we get $T(n) = O(n\beta(k)\log^2 k\log(n/k))$. The overall complexity of the algorithm `divide_and_conquer_ksp`, including sorting, is then $O(n\log n + m\log^2 k\log(n/k))$, where $m = O(n\beta(k))$ is the worst case size of a k-set polygon of n points. □

In the worst-case time complexity of the final divide and conquer algorithm appears an additional $\log(n/k)$ factor, in comparison with the algorithm of [CSY87]. This factor is suspected to come from over-estimates in the complexity computation. Indeed, in this computation, the number of edges removed and created is supposed to be at each merging step linear with the worst case sizes of the merged k-set polygons. Applied to the recursive convex hull construction (i.e. for $k = 1$), this way of computing leads to a complexity of $O(n \log n)$ after sorting whereas it has been shown in Theorem 5.8 that the algorithm is only in $O(n)$.

Chapter 6

New results on centroid triangulations

6.1 Introduction

In Chapter 3, we discovered that the number of k-sets of a convex inclusion chain of a set of points V in the plane, is equal to the number of order-k Voronoi regions of V. The purpose of this chapter is to try to understand the link between these two notions.

Recall that the order-k Voronoi diagram of V admits a dual, called the order-k Delaunay triangulation of V, whose vertices are the centroids of k-point subsets of V that define the order-k Voronoi regions. The order-k Delaunay triangulation of V forms a triangulation of the k-set polygon of V and each edge in this triangulation links the centroids of two k-point subsets of V that have $k-1$ points in common. Such a triangulation is called a k-neighbor triangulation of V.

Using the results of Chapter 4, we show first that it is possible to create a k-neighbor triangulation of V whose vertices are the centroids of the k-sets of a convex inclusion chain of V. We show next, that this k-neighbor triangulation is, as the order-k Delaunay triangulation, an order-k centroid triangulation of V; that is, it can be defined recursively from an order-$(k-1)$ triangulation of V.

By using the incremental k-set polygon construction algorithm of Chapter 4, we give an algorithm that constructs a particular order-k centroid triangulation in $O(n \log n + k(n-k) \log^2 k)$ time. The used method is close to the incremental construction of a (classical) triangulation of a set of points in the plane, where the points are processed by their increasing x-coordinates.

The order-k Delaunay triangulations remained the only triangulations that verify the recursive definition of a centroid triangulation until we found our new type of triangulations. Moreover, we know that both types of triangulations have the same number of vertices. A natural question is to know if all order-k centroid triangulations of V have the same number of vertices. As we have seen in Chapter 2, all centroid triangulations are composed of two types of triangles: Territory triangles and domain triangles. In the last section of this chapter, we show that it suffices that the domain triangles of an order-k centroid triangulation of V form convex subsets so that this triangulation has the same number of vertices as the order-k Delaunay triangulation of V, that is, $2kn - n - k^2 + 1 - \sum_{i=1}^{k-1} \gamma^i(V)$ vertices (where $\gamma^i(V)$ is the number of i-sets of V).

6.2 Convex inclusion chains and centroid triangulations

We recall first, some of the notations used in Chapter 4, and we extend them.

Let $\mathcal{V} = (v_1, \ldots, v_n)$ be a convex inclusion chain of $V = \{v_1, \ldots, v_n\}$. For every integer $i \in \{1, \ldots, n\}$, we note $V_i = \{v_1, \ldots, v_i\}$ and $\mathcal{V}_i = (v_1, \ldots, v_i)$.

From the results of Chapter 3, if $i > k+1$, the edges of $g^k(V_{i-1})$ that are not edges of $g^k(V_i)$ form a non empty polygonal line, which we will note $\mathcal{D}_i^k$ (see Figure 6.1). In the case where $i = k+1$, $g^k(V_{i-1})$ is reduced to a unique point and in this case we set $\mathcal{D}_i^k = g^k(V_{i-1})$.

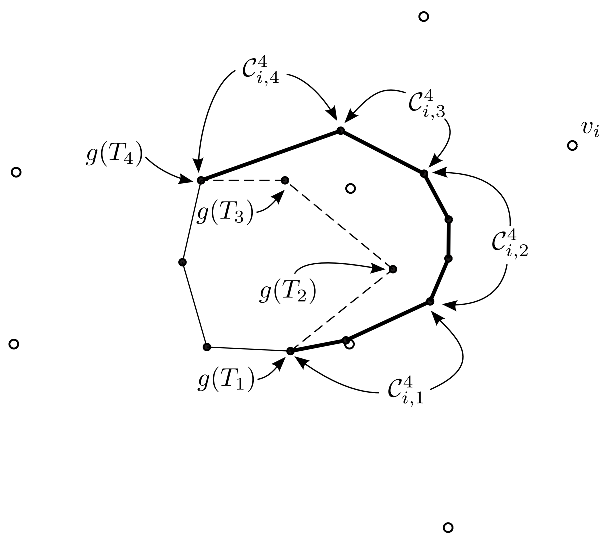

Figure 6.1: The polygonal line $\mathcal{D}_i^4 = g(T_1) \ldots g(T_4)$ and the polygonal line $\mathcal{C}_i^4 = (\mathcal{C}_{i,1}^4, \ldots \mathcal{C}_{i,4}^4)$.

In the same way, the edges of $g^k(V_i)$ that are not edges of $g^k(V_{i-1})$ form a non empty polygonal line denoted by $\mathcal{C}_i^k$. By noting $g(T_1), \ldots, g(T_m)$, the vertices of $\mathcal{D}_i^k$, $\mathcal{C}_i^k$ is decomposed into a sequence $(\mathcal{C}_{i,1}^k, \ldots, \mathcal{C}_{i,m}^k)$ of m polygonal lines that are disjoint (except at their ends) and which may be reduced to points. For every $j \in \{1, \dots, m\}$, the vertices of $\mathcal{C}^k_{i,j}$ are the vertices $g(T)$ of $\mathcal{C}^k_i$ for which there exits an oriented straight line Δ such that $g(T)$ and $g(T_j)$ are $//_\Delta$-separable from $g^k(V_{i+1})$ and of $g^k(V_i)$ respectively.

From Remark 4.9, $\mathcal{C}^k_{i,1}$ is not reduced to a point and its first edge is the unique edge of $\mathcal{C}^k_i$ of the form $e_P(s, v_i)$. In the same way, the last edge of $\mathcal{C}^k_{i,m}$ is the unique edge of $\mathcal{C}^k_i$ of the form $e_{P'}(v_i, t')$. From now on, we note by $\mathcal{C}'^k_{i,1}$ the line $\mathcal{C}^k_{i,1}$ without its first edge and by $\mathcal{C}'^k_{i,m}$ the line $\mathcal{C}^k_{i,m}$ without its last edge. For every $j \in \{2, \dots, m-1\}$, we note $\mathcal{C}'^k_{i,j} = \mathcal{C}^k_{i,j}$. Each of these lines can be reduced to a single point. The line $\mathcal{C}'^k_i = (\mathcal{C}'^k_{i,1}, \dots, \mathcal{C}'^k_{i,m})$ is then composed of the edges $e_{P''}(s'', t'')$ of $\mathcal{C}^k_i$ with $v_i \in P''$. If no such edge exists, $\mathcal{C}'^k_i$ is then reduced to a unique point (see Figure 6.2).

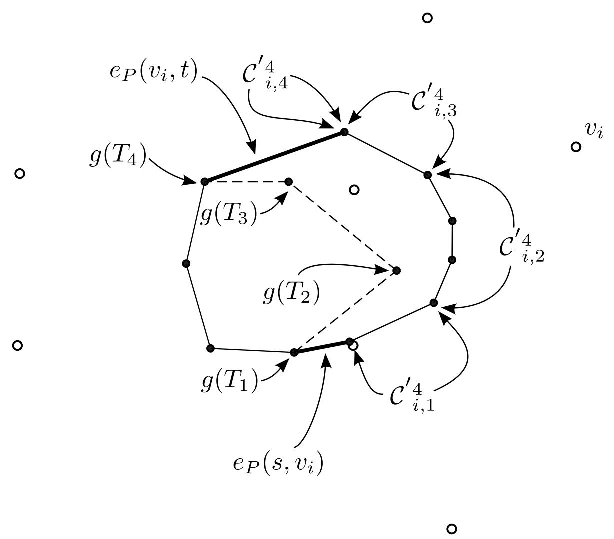

Figure 6.2: The polygonal line $\mathcal{C}'^4_i = (\mathcal{C}'^4_{i,1}, \dots \mathcal{C}'^4_{i,4})$.

Remark 6.1. *Note that from Lemma 3.5, $e_{P''}(s'', t'')$ is an edge of $\mathcal{C}'^k_i$, if and only if, $e_{P'' \setminus \{v_i\}}(s'', t'')$ is an edge of $\mathcal{D}^{k-1}_i$. Now, from Proposition 3.8, if $k > 1$, $\mathcal{D}^{k-1}_i$ contains at least one edge. Hence, the same holds for $\mathcal{C}'^k_i$.*

Lemma 6.2. *(i) For every vertex $g(T_j)$ of $\mathcal{D}^k_i$ and for every vertex $g(T)$ of $\mathcal{C}'^k_{i,j}$, there exists a point s of T_j such that $T = (T_j \setminus \{s\}) \cup \{v_i\}$.*

(ii) Moreover, the segment $g(T_j)g(T)$ is included in $\overline{g^k(V_i) \setminus g^k(V_{i-1})}$.

Proof. (i) By definition, for every vertex $g(T)$ of $\mathcal{C}'^k_{i,j}$, there exist two parallel straight lines Δ and Δ', with the same direction, such that $\mathring{\Delta}^- \cap V_{i-1} = T_j$ and $\mathring{\Delta}'^- \cap (V_{i-1} \cup \{v_i\}) = T$. Hence, since $v_i \in T$, there is exactly one point s of V_{i-1} between Δ and Δ'. It results that $T = (T_j \setminus \{s\}) \cup \{v_i\}$ (see Figure 6.3).

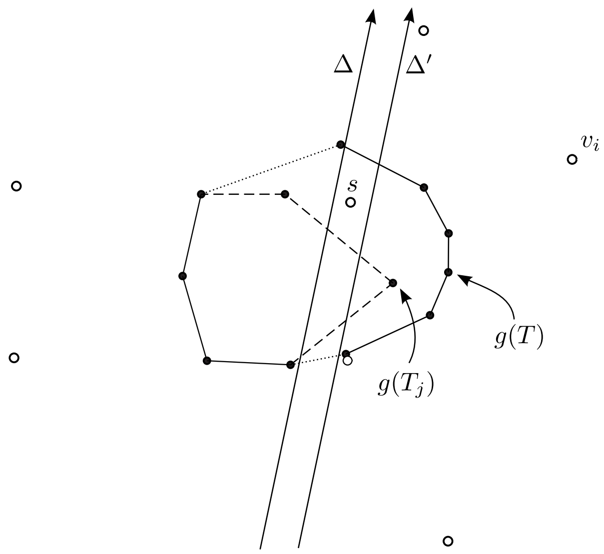

Figure 6.3: The vertices $g(T)$ and $g(T_j)$ are such that $T = (T_j \setminus \{s\}) \cup \{v_i\}$.

(ii) Let Δ'' be the straight line parallel to Δ, with the same direction as Δ and that passes through $g(T_j)$. Since $v_i \in \mathring{\Delta}'^-$ and $s \in \mathring{\Delta}'^+$, $g(T) = g((T_j \setminus \{s\}) \cup \{v_i\}) \in \mathring{\Delta}''^-$, from Lemma 2.9.

In the same way, every subset U of V_{i-1} with $|U| = k$ is of the form $U = (T_j \setminus A) \cup B$ with $A \subset \mathring{\Delta}^-$ and $B \subset \mathring{\Delta}^+$. From Lemma 2.9, we then have $g(U) \in \Delta''^+$. It results that $g^k(V_{i-1}) \subset \Delta''^+$ and that the segment $g(T_j)g(T)$ intersects $g^k(V_{i-1})$ only in $g(T_j)$. In addition, since $g(T_j)$ and $g(T)$ belong to $g^k(V_i)$, $g(T_j)g(T) \subset \overline{g^k(V_i) \setminus g^k(V_{i-1})}$ (see Figure 6.4). $\square$

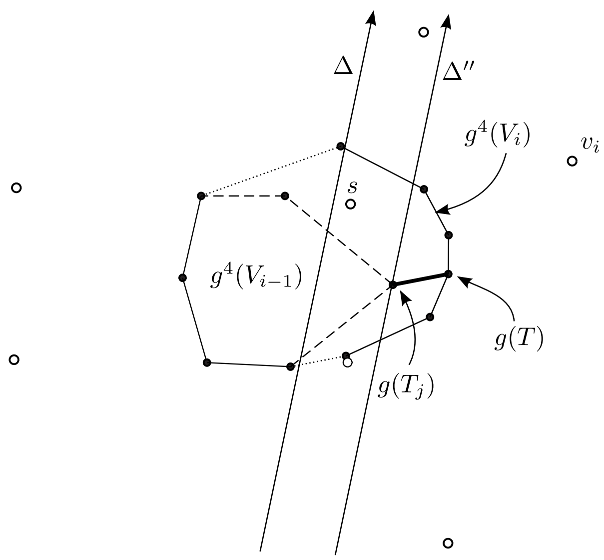

Figure 6.4: The edge $g(T_j)g(T)$ is inside $\overline{g^4(V_i) \setminus g^4(V_{i-1})}$.

From now on, we note by $\mathcal{E}_i^k$ the set of segments $g(T_j)g(T)$ obtained using the preceding lemma when j runs over $\{1,\ldots,m\}$ (see Figure 6.5).

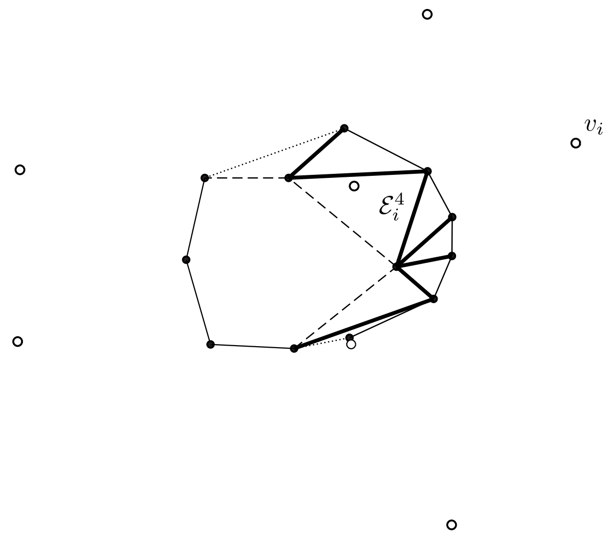

Figure 6.5: The set $\mathcal{E}_i^4$ of segments $g(T_j)g(T)$ obtained using Lemma 6.2

Remark 6.3. *By definition, the line segment that links $g(T_1)$ to the first vertex $g(T)$ of ${\mathcal{C}'}_{i,1}^k$ is the first edge of $\mathcal{C}_i^k$. If Δ is the oriented straight line spanned by the oriented edge $g(T_1)g(T)$, $g(T_1)$ is $//_\Delta$-separable from $g^k(V_{i-1})$. By slightly rotating Δ in the positive direction, we get an oriented line Δ' such that $g(T_1)$ is $//_{\Delta'}$-separable from $g^k(V_{i-1})$ and $g(T)$ is $//_{\Delta'}$-separable from $g^k(V_i)$. It results that $g(T_1)g(T)$ is an edge of $\mathcal{E}_i^k$.*

In the same way, the line segment that links the last vertex of ${\mathcal{C}'}_{i,m}^k$ to $g(T_m)$ is an edge of $\mathcal{C}_i^k$ and an edge of $\mathcal{E}_i^k$ at the same time. The boundary of $g^k(V_i)\backslash g^k(V_{i-1})$ is then made of the edges of $\mathcal{D}_i^k$, of the edges of ${\mathcal{C}'}_i^k$, and of two edges of $\mathcal{E}_i^k$. Note that, when $k=1$ and $i=2$, these two edges are geometrically the same since $\mathcal{D}_i^k$ and ${\mathcal{C}'}_i^k$ are reduced to points. $g^k(V_i)\backslash g^k(V_{i-1})$ is then reduced to a single segment.

Lemma 6.4. *(i) The line segments of $\mathcal{E}_i^k$ induce a triangulation of $\overline{g^k(V_i) \setminus g^k(V_{i-1})}$.*

(ii) The triangles of this triangulation are:

- *the triangles $g(T_j)g(T)g(T')$, where $g(T_j)$ is a vertex of $\mathcal{D}_i^k$ and where $g(T)g(T')$ is an edge of ${\mathcal{C}'}_{i,j}^k$,*
- *the triangles $g(T_j)g(T_{j+1})g(T)$, where $g(T_j)g(T_{j+1})$ is an edge of $\mathcal{D}_i^k$ and where $g(T)$ is the common vertex of ${\mathcal{C}'}_{i,j}^k$ and ${\mathcal{C}'}_{i,j+1}^k$.*

The first triangles are domain triangles and the second ones are territory triangles.

Proof. (i.1) Let us show first that the segments of $\mathcal{E}_i^k$ are pairwise disjoint (except at their endpoints). Let $g(T_j)$ and $g(T_{j'})$ be two vertices of $\mathcal{D}_i^k$, $g(T)$ a vertex of ${\mathcal{C}'}_{i,j}^k$, and $g(T')$ a vertex of ${\mathcal{C}'}_{i,j'}^k$ such that we do not have $j = j'$ and $T = T'$ simultaneously.

If $j = j'$, $g(T)$ and $g(T')$ are two distinct vertices of $g^k(V_i)$. Therefore, $g(T_j)\overset{\circ}{)}g(T)$ and $g(T_{j'})\overset{\circ}{)}g(T')$ are disjoint.

If $j \neq j'$, we can suppose that, within a permutation of j and j', $j < j'$, *i.e.* $g(T_j)$ precedes $g(T_{j'})$ on $\mathcal{D}_i^k$. In this case, from Corollary 4.6, $\mathcal{C}_{i,j}^k$ precedes $\mathcal{C}_{i,j'}^k$ on $\mathcal{C}_i^k$ and both of these lines have at most one common vertex. Since ${\mathcal{C}'}_{i,j}^k \subseteq \mathcal{C}_{i,j}^k$ and ${\mathcal{C}'}_{i,j'}^k \subseteq \mathcal{C}_{i,j'}^k$, it results that $g(T_j)\overset{\circ}{)}g(T)$ and $g(T_{j'})\overset{\circ}{)}g(T')$ are disjoint.

(i.2) When $g^k(V_i) \setminus g^k(V_{i-1})$ is reduced to a single segment, this segment belongs to $\mathcal{E}_i^k$, from Remark 6.3. Otherwise, the boundary of $g^k(V_i) \setminus g^k(V_{i-1})$ is made of the edges of $\mathcal{D}_i^k$ and of ${\mathcal{C}'}_i^k$ and of two edges of $\mathcal{E}_i^k$. Since every line segment of $\mathcal{E}_i^k$ links a point of $\mathcal{D}_i^k$ to a point of ${\mathcal{C}'}_i^k$, the boundary Γ of every connected component of $(g^k\overset{\circ}{(}V_i) \setminus g^k(V_{i-1})) \setminus \mathcal{E}_i^k$ is also composed of edges of $\mathcal{D}_i^k$, of ${\mathcal{C}'}_i^k$ and of exactly two edges of $\mathcal{E}_i^k$. If Γ contains an edge $g(T)g(T')$ of ${\mathcal{C}'}_i^k$, then from Corollary 4.6, there exists one and only one vertex $g(T_j)$ of $\mathcal{D}_i^k$ such that $g(T)g(T')$ is an edge of ${\mathcal{C}'}_{i,j}^k$. By definition, the edges $g(T)g(T_j)$ and $g(T')g(T_j)$ belong then to $\mathcal{E}_i^k$ and Γ is the triangle $g(T)g(T')g(T_j)$. In the same way, if Γ contains an edge $g(T_j)g(T_{j+1})$ of $\mathcal{D}_i^k$, then from Corollary 4.6, $\mathcal{C}_{i,j}^k$ and $\mathcal{C}_{i,j+1}^k$ have a vertex $g(T)$ in common. By definition, $g(T_j)g(T)$ and $g(T_{j+1})g(T)$ are also line segments of $\mathcal{E}_i^k$ then, and Γ is the triangle $g(T_j)g(T_{j+1})g(T)$.

It results that every connected component of $(g^k\overset{\circ}{(}V_i)\setminus g^k(V_{i-1}))\setminus\mathcal{E}_i^k$ is a triangle and therefore, $\mathcal{E}_i^k$ induces a triangulation of $\overline{g^k(V_i) \setminus g^k(V_{i-1})}$.

(ii) From the proof of (i.2), the triangulation induced by $\mathcal{E}_i^k$ admits exactly two types of triangles: The first type of triangles is of the form $\gamma = g(T)g(T')g(T_j)$, where $g(T)g(T')$ is an edge of ${\mathcal{C}'}_{i,j}^k$. From Lemma 6.2, there exist two points s and s' of T_j such that $T = (T_j \setminus \{s\}) \cup \{v_i\}$ and $T' = (T_j \setminus \{s'\}) \cup \{v_i\}$. By setting $P = T_j \setminus \{s, s'\}$, γ is the domain triangle $g(P \cup \{s, v_i\})g(P \cup \{s', v_i\})g(P \cup \{s, s'\})$.

The second type of triangles is of the form $\gamma = g(T_j)g(T_{j+1})g(T)$, where $g(T)$ is the common vertex of ${\mathcal{C}'}_{i,j}^k$ and ${\mathcal{C}'}_{i,j+1}^k$. From Lemma 6.2, there exist $s \in T_j$ and $s' \in T_{j+1}$ such that $T_j = (T \setminus \{v_i\}) \cup \{s\}$ and $T_{j+1} = (T \setminus \{v_i\}) \cup \{s'\}$. By setting $P = T \setminus \{v_i\}$, γ is the territory triangle $g(P \cup \{s\})g(P \cup \{s'\})g(P \cup \{v_i\})$ (see Figure 6.6). $\square$

Proposition 6.5. *For all the integers i of $\{k+1, \ldots, n\}$, the set of edges of the k-set polygons $g^k(V_i)$ and of the sets $\mathcal{E}_i^k$ form a k-neighbor triangulation of V whose vertices are determined by the set of k-sets of the convex inclusion chain $(v_1, \ldots, v_n)$.*

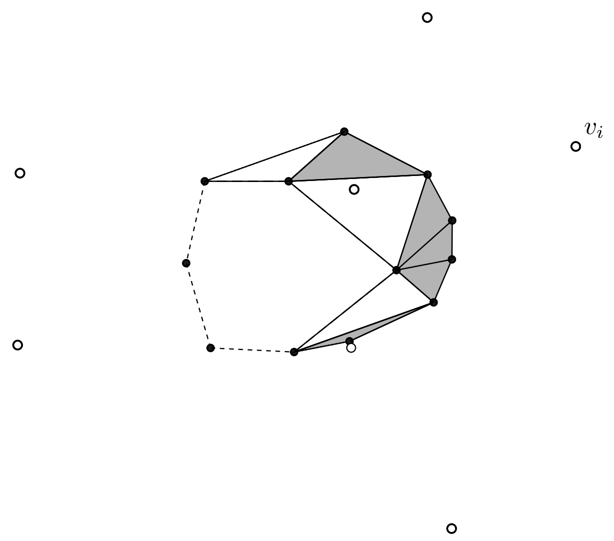

Figure 6.6: The domain triangles (in gray) and the territory triangles (in white) once $\overline{g^4(V_i) \setminus g^4(V_{i-1})}$ is triangulated

Proof. The k-set polygon $g^k(V_k)$ is reduced to a unique point. From Lemma 6.4, if $i \in \{k+1, \dots, n\}$, $\mathcal{E}_i^k$ induces a triangulation of $\overline{g^k(V_i) \setminus g^k(V_{i-1})}$. It results that, when i runs over $\{k+1, \dots, n\}$, the set of edges of all the $g^k(V_i)$ and of all the $\mathcal{E}_i^k$ forms a triangulation $\mathcal{T}$ of $g^k(V_n) = g^k(V)$ (see Figure 6.7).

Moreover, from Proposition 2.12, every edge of $g^k(V_i)$ is of the form $g(T)g(T')$, with $|T \cap T'| = k - 1$. From Lemma 6.2, the same holds for the edges of $\mathcal{E}_i^k$.

Since the vertices of $\mathcal{T}$ are the vertices of all the k-set polygons $g^k(V_i)$, for $i \in \{k, \dots, n\}$, then from Lemma 3.11, these vertices are the centroids of the k-sets of the convex inclusion chain $(v_1, \dots, v_n)$. Also from Lemma 3.11, these centroids are pairwise disjoint. It results that $\mathcal{T}$ is a k-neighbor triangulation of V whose vertices are determined by the k-sets of $(v_1, \dots, v_n)$. □

For every convex inclusion chain $\mathcal{V}$ of V, the triangulation defined by Proposition 6.5 is noted by $\mathcal{T}^k(\mathcal{V})$. In the particular case where $k = n$, we note $\mathcal{T}^n(\mathcal{V}) = g^n(V) = g(V)$.

We show now that $\mathcal{T}^k(\mathcal{V})$ is an order-k triangulation of V as defined in Chapter 2, that is, it can be obtained from a triangulation of V after $k-1$ applications of the order-k triangulation construction algorithm. For every set V of n points, we call sequence of centroid triangulations of V, every sequence $(\mathcal{A}^1, \dots, \mathcal{A}^n)$ of centroid triangulations of V such that $\mathcal{A}^1$ is a triangulation of V and, for every integer $i \in \{2, \dots, n\}$, $\mathcal{A}^i$ is obtained from $\mathcal{A}^{i-1}$ using the order-k triangulation construction algorithm. Recall that $\mathcal{A}^n = g^n(V)$ is reduced to the unique point $g(V)$.

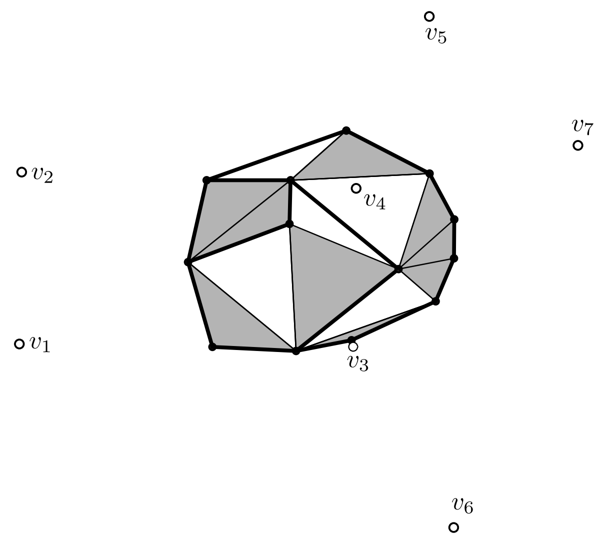

Figure 6.7: The 4-neighbor triangulation determined by the k-sets of the convex inclusion chain $(v_1, \ldots, v_7)$

Theorem 6.6. *For every convex inclusion chain $\mathcal{V} = (v_1, \ldots, v_n)$ of V, $(\mathcal{T}^1(\mathcal{V}), \ldots, \mathcal{T}^n(\mathcal{V}))$ is a sequence of centroid triangulations of V.*

Proof. The set $V_1 = \{v_1\}$ admits a unique convex inclusion chain $\mathcal{V}_1 = (v_1)$. The triangulation $\mathcal{T}^1(\mathcal{V}_1)$ is reduced to the point v_1 and $(\mathcal{T}^1(\mathcal{V}_1))$ is an (elementary) sequence of centroid triangulations of V_1.

Now, let i be an integer of $\{2, \ldots, n\}$ and $\mathcal{V}_{i-1}$ be the convex inclusion chain $(v_1, \ldots, v_{i-1})$ of $V_{i-1} = \{v_1, \ldots, v_{i-1}\}$. Suppose that the following induction hypothesis holds: $(\mathcal{T}^1(\mathcal{V}_{i-1}), \ldots, \mathcal{T}^{i-1}(\mathcal{V}_{i-1}))$ is a sequence of centroid triangulations of V_{i-1}. We show now that if $\mathcal{V}_i$ is the convex inclusion chain $(v_1, \ldots, v_i)$ of $V_i = \{v_1, \ldots, v_i\}$ then $(\mathcal{T}^1(\mathcal{V}_i), \ldots, \mathcal{T}^i(\mathcal{V}_i))$ is also a sequence of centroid triangulations of V_i. From Proposition 6.5, $\mathcal{T}^1(\mathcal{V}_i)$ is a triangulation of V_i and is then the first element of a centroid triangulation sequence of V_i. Suppose also that this second induction hypothesis holds: For an integer $k \leq i$, $(\mathcal{T}^1(\mathcal{V}_i), \ldots, \mathcal{T}^{k-1}(\mathcal{V}_i))$ is the beginning of a centroid triangulation sequence of V_i.

If $k = i$, by definition, $\mathcal{T}^k(\mathcal{V}_i)$ is reduced to the unique point $g^k(V) = g(V)$. Therefore, $(\mathcal{T}^1(\mathcal{V}_i), \ldots, \mathcal{T}^k(\mathcal{V}_i))$ is a centroid triangulation sequence of V_i.

We show now, that when $k < i$, $\mathcal{T}^k(\mathcal{V}_i)$ can be obtained from $\mathcal{T}^{k-1}(\mathcal{V}_i)$ using the order-k triangulation construction algorithm.

By construction, $\mathcal{T}^k(\mathcal{V}_i)$ is the union of $\mathcal{T}^k(\mathcal{V}_{i-1})$ and of $\overline{\mathcal{T}^k(\mathcal{V}_i) \setminus \mathcal{T}^k(\mathcal{V}_{i-1})}$.

(i) From the first induction basis, $\mathcal{T}^k(\mathcal{V}_{i-1})$ is obtained from $\mathcal{T}^{k-1}(\mathcal{V}_{i-1})$ using the order-k triangulation construction algorithm. In particular, the domain triangles of $\mathcal{T}^k(\mathcal{V}_{i-1})$ are obtained from the territory triangles of $\mathcal{T}^{k-1}(\mathcal{V}_{i-1})$ by the first step of the order-k triangulation construction algorithm.

(ii) From Lemma 6.4, every domain triangle of $\overline{\mathcal{T}^k(\mathcal{V}_i) \setminus \mathcal{T}^k(\mathcal{V}_{i-1})}$ has one and only one edge on ${\mathcal{C}'}_i^k$. Conversely, every edge of ${\mathcal{C}'}_i^k$ is an edge of such a triangle. Moreover, from Lemma 6.2, these triangles are of the form $g(P \cup \{s, s'\})g(P \cup \{v_i, s\})g(P \cup \{v_i, s'\})$, where $g(P \cup \{v_i, s\})g(P \cup \{v_i, s'\})$ is the edge on ${\mathcal{C}'}_i^k$. Now, as recalled in Remark 6.1, $g(P \cup \{v_i, s\})g(P \cup \{v_i, s'\})$ is an edge of ${\mathcal{C}'}_i^k$ if, and only if, $g(P \cup \{s\})g(P \cup \{s'\})$ is an edge of $\mathcal{D}_i^{k-1}$. Moreover, from Lemma 6.4, every territory triangle of $\overline{\mathcal{T}^{k-1}(\mathcal{V}_i) \setminus \mathcal{T}^{k-1}(\mathcal{V}_{i-1})}$ has one, and only one, edge on $\mathcal{D}_i^{k-1}$, and every edge of $\mathcal{D}_i^{k-1}$ is an edge of such a triangle. From Lemma 6.2, $g(P \cup \{v_i\})$ is then the third vertex of the territory triangle that has $g(P \cup \{s\})g(P \cup \{s'\})$ as an edge. It results that $g(P \cup \{s, s'\})g(P \cup \{v_i, s\})g(P \cup \{v_i, s'\})$ is a domain triangle of $\overline{\mathcal{T}^k(\mathcal{V}_i) \setminus \mathcal{T}^k(\mathcal{V}_{i-1})}$ if, and only if, $g(P \cup \{s\})g(P \cup \{s'\})g(P \cup \{v_i\})$ is a territory triangle of $\overline{\mathcal{T}^{k-1}(\mathcal{V}_i) \setminus \mathcal{T}^{k-1}(\mathcal{V}_{i-1})}$. The domain triangles of $\overline{\mathcal{T}^k(\mathcal{V}_i) \setminus \mathcal{T}^k(\mathcal{V}_{i-1})}$ can be obtained then from the territory triangles of $\overline{\mathcal{T}^{k-1}(\mathcal{V}_i) \setminus \mathcal{T}^{k-1}(\mathcal{V}_{i-1})}$ in the first step of the order-k triangulation construction algorithm.

(iii) It results from (i) and (ii) that all the domain triangles of $\mathcal{T}^k(\mathcal{V}_i)$ are obtained from the territory triangles of $\mathcal{T}^{k-1}(\mathcal{V}_i)$ in the first step of the order-k triangulation construction algorithm.

(iv) Let us show now that each vertex of $\mathcal{T}^k(\mathcal{V}_i)$ is a vertex of a domain triangle of $\mathcal{T}^k(\mathcal{V}_i)$. By construction, the vertices of $\mathcal{T}^k(\mathcal{V}_i)$ are the vertex $g(V_k)$ and, for every $h \in \{k+1, \dots, i\}$, the vertices of ${\mathcal{C}'}_h^k$. Since $g(V_k)$ is also a vertex of $\mathcal{T}^k(\mathcal{V}_{k+1})$ and since all the triangles of $\mathcal{T}^k(\mathcal{V}_{k+1})$ are domain triangles from Lemma 6.2, then $g(V_k)$ is a vertex of a domain triangle of $\mathcal{T}^k(\mathcal{V}_i)$. From Remark 6.1, for every $h \in \{k+1, \dots, i\}$, ${\mathcal{C}'}_h^k$ is not reduced to a unique point. Every vertex of ${\mathcal{C}'}_h^k$ is then the endpoint of an edge of ${\mathcal{C}'}_h^k$. It follows, from Lemma 6.4, that every vertex of ${\mathcal{C}'}_h^k$ is a vertex of a domain triangle of $\mathcal{T}^k(\mathcal{V}_h)$ and, therefore, of $\mathcal{T}^k(\mathcal{V}_i)$.

(v) By noting τ, the set of domain triangles of $\mathcal{T}^k(\mathcal{V}_i)$, the territory triangles of $\mathcal{T}^k(\mathcal{V}_i)$ form then a constrained triangulation of $\overline{g^k(V_i) \setminus \tau}$ and can be obtained using the second step of the order-k triangulation construction algorithm.

(vi) It results from (iii) and from (v) that $\mathcal{T}^k(\mathcal{V}_i)$ can be obtained from $\mathcal{T}^k(\mathcal{V}_{i-1})$ using the order-k triangulation construction algorithm. It follows, from the second induction hypothesis that $(\mathcal{T}^1(\mathcal{V}_i), \dots, \mathcal{T}^k(\mathcal{V}_i))$ is the beginning of a centroid triangulation sequence of V_i, for every $k \leq i$. It results that $(\mathcal{T}^1(\mathcal{V}_i), \dots, \mathcal{T}^i(\mathcal{V}_i))$ is a centroid triangulation sequence of V_i, for every $i \in \{2, \dots, n\}$. $(\mathcal{T}^1(\mathcal{V}), \dots, \mathcal{T}^n(\mathcal{V}))$ is then a centroid triangulation sequence of V. $\square$

6.3 Construction of a centroid triangulation

Now we give an algorithm and the associated data structure for building the centroid triangulation $\mathcal{T}^k(\mathcal{V})$, for some particular convex inclusion chain $\mathcal{V}$. This algorithm is an extended version of the algorithm given in Chapter 4.

6.3.1 Data structure

To store a centroid triangulation we will use a map as data structure. This map is a graph where the edges are ordered around each vertex in a circular list based on the trigonometric order they appear in.

Similarly to Chapter 4, every edge in this triangulation is of the form $g(P \cup \{s\})g(P \cup \{t\})$. It suffices to store in each edge in the data structure references to the points s and t. Also the set T of one vertex $g(T)$ of this triangulation is stored and all the other vertices can be found when walking along the edges of this triangulation, starting at the vertex $g(T)$. Here we will chose for $g(T)$ a vertex of the boundary of the triangulation.

Clearly, the algorithms of Chapter 4, can be adapted easily to work with k-set polygons stored using this map.

6.3.2 Algorithm

As in Chapter 4, let $\mathcal{V} = (v_1, \ldots, v_n)$ be a convex inclusion chain of V that forms a simple polygonal line. The functions `k_set_polygon::initialize(`$v_1, \ldots, v_{k+1}$`)` and `k_set_polygon::build_ci(`v_i`)` of Chapter 4 can be used as such to construct the subset of the edges of $\mathcal{T}^k(\mathcal{V})$ that are also the edges of the k-set polygons $g^k(V_i)$, $i \in \{k+1, \ldots, n\}$. It remains to show how to construct the edges $\mathcal{E}_i^k$, for all $i \in \{k+1, \ldots, n\}$.

Since $g^k(V_k)$ is reduced to the unique vertex $g(V_k)$, which is also a vertex of $g^k(V_{k+1})$, constructing $\mathcal{E}_{k+1}^k$ comes from Lemma 6.2, to triangulate $g^k(V_{k+1})$ by connecting $g(V_k)$ to all the other vertices of $g^k(V_{k+1})$. This can be done while constructing $g^k(V_{k+1})$ in function `initialize(`$v_1, \ldots, v_{k+1}$`)`. For all $i \in \{k+2, \ldots, n\}$, the edges of $\mathcal{E}_i^k$ link one vertex of $\mathcal{C'}_i^k$ to a vertex of $\mathcal{D}_i^k$. More precisely, the edges of $\mathcal{E}_i^k$ are the edges that link the vertices of $\mathcal{C'}_{i,j}^k$ to the vertex $g(T_j)$ of $\mathcal{D}_i^k$, when j runs over $\{1, \ldots, m\}$.

Now, the set T_j and the vertex $g(T_j)$ precisely are used by the function `build_ci(`v_i`)` to build the line part $\mathcal{C'}_{i,j}^k$ of $\mathcal{C}_i^k$. It suffices then, while building each vertex of $\mathcal{C'}_{i,j}^k$ in the function `build_ci(`v_i`)`, to create a link between this vertex and the vertex $g(T_j)$.

Theorem 6.7. *The given algorithm constructs $\mathcal{T}^k(\mathcal{V})$ in $O(k(n-k)\log^2 k)$ time.*

Proof. The edges of $\mathcal{T}^k(\mathcal{V})$ that are edges of the k-set polygons $g^k(V_i)$, $i \in \{k+2,\dots,n\}$, are built by the algorithm `k_set_polygon::build` whose complexity is in $O(k(n-k)\log^2 k)$, from Theorem 4.16.

For each created vertex of a k-set polygon $g^k(V_i)$, $i \in \{k+1,\dots,n\}$, one and only one edge of $\mathcal{E}_i^k$ is created in $O(1)$ time. The construction time for all the edges of $\mathcal{E}_i^k$ is then proportional to the number of k-sets of the convex inclusion chain $\mathcal{V}$, that is $O(k(n-k))$ from Theorem 3.12. □

Given a set V of n points, a particular convex inclusion chain of V can be obtained by sorting all the points of V according to their x-coordinates in the increasing order. It then results:

Corollary 6.8. *Given a set V of n points, a particular order-k centroid triangulation of V can be obtained in $O(n\log n + k(n-k)\log^2 k)$ time.*

6.4 Centroid triangulation size

Let V be a set of n points and let $(\mathcal{T}^1,\dots,\mathcal{T}^n)$ be a centroid triangulation sequence of V.

If, for some subset T of $k+1 \geq 2$ points of V, the set composed of the edges of $\mathcal{T}^k$ of the form $g(T\setminus\{s\})g(T\setminus\{t\})$ and of the domain triangles of $\mathcal{T}^k$ of the form $g(T\setminus\{r\})g(T\setminus\{s\})g(T\setminus\{t\})$ is not empty, then this set is called the domain of T in $\mathcal{T}^k$.

Note that every edge and every domain triangle of $\mathcal{T}^k$ belongs to one and only one domain, since a unique set of k points is associated to each vertex of $\mathcal{T}^k$. Moreover, the edges of a domain triangle belong to the same domain as the triangle. We claim now the following conjecture:

Domain convexity conjecture. *Every domain is either reduced to a line segment or is a triangulation of a convex polygon.*

This conjecture results from various experimentations that we have done. Note that the result of this conjecture has already been proved in the case of order-k Delaunay triangulations [Sch95].

We show now that this result is sufficient to find the size of the centroid triangulations.

Lemma 6.9. *For every integer $k \in \{2,\dots,n\}$, if the subset T of k points admits a domain in $\mathcal{T}^{k-1}$ then $g(T)$ is a vertex of $\mathcal{T}^k$.*

Proof. In the particular case where $k = n$, from Property 2.22, $\mathcal{T}^{k-1}$ is made of a unique domain and this domain is necessarily the domain of V. On the other hand, $g(V)$ is the unique vertex of $\mathcal{T}^k = g^n(V)$.

We deal now with the general case where $k \in \{2, \ldots, n-1\}$.

(i) Let us show first that $\mathcal{T}^{k-1}$ admits at least two distinct domains.

Let $e_P(s,t)$ be an edge of $g^{k-1}(V)$. Since $|(\overset{\circ}{st})^- \cap V| = |P| = k-2$ and since $k < n$, $(st)^+$ contains at least one point of V. Moreover, at least one of the points of $V \cap (st)^+$ is a vertex of $conv(V)$. Let r be such a point. There exists then a point q of $V \setminus \{r\}$ such that $|V \cap (rq)^-| = k-2$. It follows that $e_{V\cap(rq)^-}(r,q)$ is an edge of $g^{k-1}(V)$. Since $r \notin P$, this edge does not belong to the same domain as $e_P(s,t)$.

(ii) Let T be a subset of k points of V that admits a domain in $\mathcal{T}^{k-1}$. Since the edges of a domain triangle belong to the same domain as the triangle and since every edge belongs to a unique domain, an edge cannot be common to two triangles belonging to two distinct domains. Moreover, from (i), $\mathcal{T}^{k-1}$ admits at least two distinct domains. It results that the domain of T admits at least one edge that is also an edge of a territory triangle of $\mathcal{T}^{k-1}$. This edge is of the form $g(T \setminus \{s\})g(T \setminus \{t\})$ and the third vertex of the territory triangle is of the form $g(T \setminus \{s,t\} \cup \{r\})$. By applying the order-$k$ triangulation construction algorithm, $g(T)g((T \setminus \{t\}) \cup \{r\})g((T \setminus \{s\}) \cup \{r\})$ is then a domain triangle of $\mathcal{T}^k$. $g(T)$ is then a vertex of $\mathcal{T}^k$. □

Lemma 6.10. *If the domain convexity conjecture is verified then, for every $k \in \{2, \ldots, n\}$, if $g(T)$ is an extreme point of $g^k(V)$ then T admits a domain in $\mathcal{T}^{k-1}$.*

Proof. (i) For every $l \in \{1, \ldots, k-1\}$, let $\mathcal{G}^l$ be the subgraph of $\mathcal{T}^l$ whose vertices are the centroids of subsets of l points of T. Let us show first that $\mathcal{G}^l$ is connected (see Figures 6.8 and 6.9).

Since $g(T)$ is an extreme point of $g^k(V)$, there exists a straight line Δ that strictly separates T from $V \setminus T$. Within a rotation of V, we can suppose that Δ is horizontal, is not parallel to any straight line that passes through two points of V, and that T is above Δ. If P is the set of l points of T of maximal ordinates, then from Lemma 2.10, $g(P)$ is the unique vertex of $g^l(V)$ with maximal ordinate. Therefore, it is also the unique vertex of maximal ordinate of $\mathcal{G}^l$. Every other vertex $g(R)$ of $\mathcal{G}^l$ admits then at least one neighbor $g(R')$ in $\mathcal{T}^l$ whose ordinate is greater than the one of $g(R)$. From the properties of the edges of $\mathcal{T}^l$, there exist $r \in R \setminus R'$ and $r' \in R' \setminus R$ such that $R' = (R \setminus \{r\}) \cup \{r'\}$ and so $g(R)g(R')$ is the image of rr' by an homothety of ratio $1/l$. Hence, the abscissa of r' is greater than the abscissa of r, which proves that $g(R')$ is also a vertex of $\mathcal{G}^l$. $\mathcal{G}^l$ is then a connected graph.

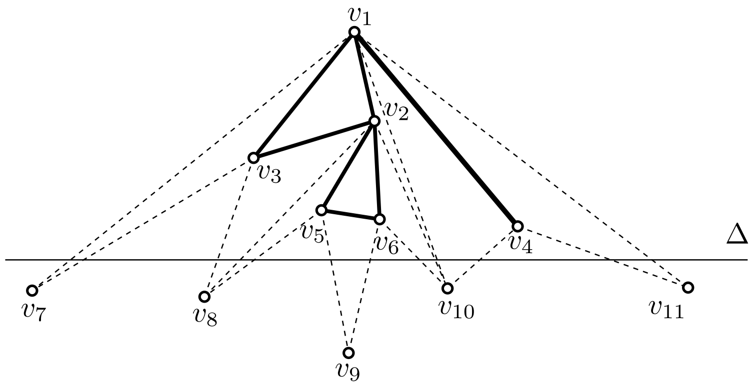

Figure 6.8: A subset of $\mathcal{T}^1$ with $k = 6$ and $T = \{1, \ldots, 6\}$. The graph $\mathcal{G}^1$ (full lines) admits 6 vertices, 7 edges, and 2 territory triangles

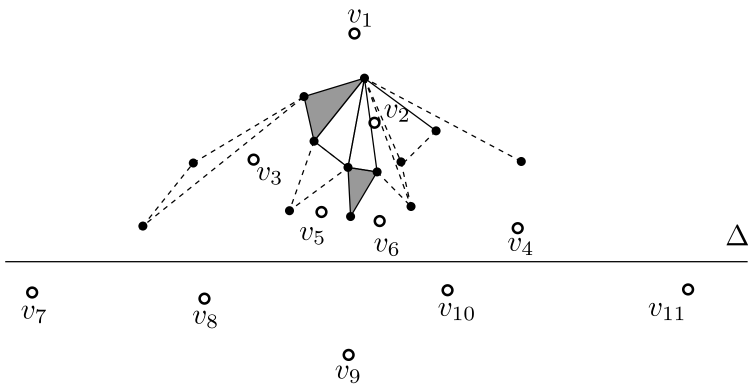

Figure 6.9: A subset of an order-2 triangulation $\mathcal{T}^2$ corresponding to the triangulation $\mathcal{T}^1$ of Figure 6.8. The graph $\mathcal{G}^2$ (full lines) contains 8 vertices, 10 edges, 2 domain triangles (gray) corresponding to the territory triangles of $\mathcal{T}^1$, and 2 territory triangles (white)

(ii) Let us show now by contradiction that every face of $\mathcal{G}^l$ is a face of $\mathcal{T}^l$. Suppose that Γ is a subset of the plane bounded by a face of $\mathcal{G}^l$ that is not a triangle of $\mathcal{T}^l$. There exists then an edge of $\mathcal{T}^l$ that is not an edge of $\mathcal{G}^l$, that has a vertex $g(T)$ of Γ as an end point, and that is inside Γ. The other endpoint $g(Q_1)$ of this edge does not belong then to $\mathcal{G}^l$ and is inside Γ. Now, every vertex of $\mathcal{T}^l$ other than the vertices of minimal ordinate, admits a neighbor in $\mathcal{T}^l$ whose ordinate is less than its own. It results that there exists a path $L = (g(Q_1), g(Q_2), \ldots, g(Q_m))$ that links $g(Q_1)$ to one of the vertices $g(Q_m)$ of $\mathcal{T}^l$ with minimal ordinate and such that, $\forall i \in \{2, \ldots, m\}$, the ordinate of $g(Q_i)$ is less than the one of $g(Q_{i-1})$. Since two edges of $\mathcal{T}^l$ can only intersect at their endpoints, Γ and L have one vertex $g(Q_i)$ in common, with $i \in \{2, \ldots, m\}$. This vertex belongs to $\mathcal{G}^l$ as all the vertices of Γ. Since $g(Q_{i-1})$ has a greater abscissa than $g(Q_i)$, it results from the proof of (i) that $g(Q_{i-1})$ belongs also to $\mathcal{G}^l$. In the same way, $g(Q_{i-2}), \ldots, g(Q_1)$ belong to $\mathcal{G}^l$, which is impossible. It results that all the faces of $\mathcal{G}^l$ are triangles of $\mathcal{T}^l$.

(iii) We show now that $\mathcal{G}^{k-1}$ contains at least one edge. For every $l \in \{1, \ldots, k\}$, let g^l, e^l, t^l_{ter}, and t^l_{dom} be the respective numbers of vertices, of edges, of territory triangles, and of domain triangles of $\mathcal{T}^l$ that are in $\mathcal{G}^l$. Since $\mathcal{G}^l$ is a planar graph and since, from (i), all of its faces are triangles of $\mathcal{T}^l$, we have from Euler's relation:

$$g^l = e^l - (t^l_{ter} + t^l_{dom}) + 1 \tag{6.1}$$

If P is a subset of $l+1$ points of T that admits a domain in $\mathcal{T}^l$ then, from the definition of domains and of $\mathcal{G}^l$, the edges and triangles of the domain of P are edges and triangles of $\mathcal{G}^l$. Let e^l_P and t^l_P be the numbers of these edges and triangles. Since the domain of T is either reduced to a line segment or is a triangulation of a convex polygon from the domain convexity conjecture, we have for every $l \in \{1, \ldots, k-1\}$,

$$e^l_P = 2t^l_P + 1$$

Now, every edge and every domain triangle of $\mathcal{T}^l$ belongs to one, and only one, domain. By summing the previous relation over the set of domains of $\mathcal{T}^l$ that belong to $\mathcal{G}^l$, we find that the number of domains in $\mathcal{G}^l$ is equal to $e^l - 2t^l_{dom}$. Now, from Lemma 6.9, if P admits a domain in $\mathcal{G}^l$ then $g(P)$ is a vertex of $\mathcal{T}^{l+1}$ and, since $P \subseteq T$, $g(P)$ is also a vertex of $\mathcal{G}^{l+1}$. Hence:

$$g^{l+1} \geq e^l - 2t^l_{dom} \tag{6.2}$$

From (6.1) and (6.2) it results that:

$$g^{l+1} \geq g^l + t^l_{ter} - t^l_{dom} - 1$$

From Conjecture 2.25, there exists a bijection between the territory triangles of $\mathcal{T}^l$ and the domain triangles of $\mathcal{T}^{l+1}$. Moreover, from the definition of $\mathcal{G}^l$, a territory triangle belongs to $\mathcal{G}^l$ if, and only if, the domain triangle to which it is associated belongs to $\mathcal{G}^{l+1}$. Hence, $t^l_{ter} = t^{l+1}_{dom}$ and the previous relation becomes, for every $l \in \{1, \ldots, k-1\}$,

$$g^{l+1} \geq g^l + t^{l+1}_{dom} - 1$$

We then have:

$$\begin{aligned} g^{k-1} &\geq g^{k-2} + t^{k-1}_{dom} - t^{k-2}_{dom} - 1 \\ g^{k-2} &\geq g^{k-3} + t^{k-2}_{dom} - t^{k-3}_{dom} - 1 \\ &\vdots \\ g^2 &\geq g^1 + t^2_{dom} - t^1_{dom} - 1 \end{aligned}$$

By summing these equations we obtain:

$$g^{k-1} \geq t_{dom}^{k-1} + g^1 - t_{dom}^1 - (k-2)$$

Since $\mathcal{T}^1$ is a triangulation of V, g^1 is the number k of points of T. Moreover, from Property 2.22, $\mathcal{T}^1$ does not admit any domain triangle, *i.e.*, $t_{dom}^1 = 0$. Hence:

$$g^{k-1} \geq t_{dom}^{k-1} + 2 \geq 2$$

The graph $\mathcal{G}^{k-1}$ admits then at least two vertices and, since it is connected from (i), it admits at least one edge. From the definition of $\mathcal{G}^{k-1}$, this edge is of the form $g(T \setminus \{s\})g(T \setminus \{t\})$ and is thus an edge of the domain of T in $\mathcal{T}^{k-1}$. □

Proposition 6.11. *If the domain convexity conjecture is verified then, for every $k \in \{2, \ldots, n\}$, $g(T)$ is a vertex of $\mathcal{T}^k$ if, and only if, T admits a domain in $\mathcal{T}^{k-1}$.*

Proof. (i) From Lemma 6.10, if $g(T)$ is an extreme point of $g^k(V)$ then T admits a domain in $\mathcal{T}^{k-1}$. By construction, every other vertex $g(T)$ of $\mathcal{T}^k$ is obtained in the first step of the order-k triangulation construction algorithm. $g(T)$ is then the vertex of a domain triangle of $\mathcal{T}^k$ of the form $g(T)g((T \setminus \{t\}) \cup \{r\})g((T \setminus \{s\}) \cup \{r\})$ that has been generated from a territory triangle $g(T \setminus \{s\})g(T \setminus \{t\})g((T \setminus \{s, t\}) \cup \{r\})$ of $\mathcal{T}^{k-1}$. It results that the domain of T contains at least the edge $g(T \setminus \{s\})g(T \setminus \{t\})$ in $\mathcal{T}^{k-1}$.

(ii) Conversely, from Lemma 6.9, if T admits a domain in $\mathcal{T}^{k-1}$ then $g(T)$ is a vertex of $\mathcal{T}^k$. □

Theorem 6.12. *If the domain convexity conjecture is verified then, for every $k \in \{1, \ldots, n-1\}$, every order-k triangulation of V admits $2kn - n - k^2 + 1 - \sum_{i=1}^{k-1} \gamma^i(V)$ vertices (where $\gamma^i(V)$ is the number of i-sets of V and where $\sum_{i=1}^{0} = 0$).*

Proof. Let $(\mathcal{T}^1, \ldots, \mathcal{T}^n)$ be a sequence of centroid triangulations of V and, for every $k \in \{1, \ldots, n\}$, let g^k, e^k, t^k, t_{ter}^k, and t_{dom}^k the numbers of vertices, of edges, of triangles, of territory triangles, and of domain triangles of $\mathcal{T}^k$ respectively. Since, for every $k \in \{1, \ldots, n-1\}$, $\mathcal{T}^k$ is a triangulation of $g^k(V)$, each edge of $\mathcal{T}^k$ that is (resp. is not) an edge of $g^k(V)$ is adjacent to exactly two (resp. one) triangle(s) of $\mathcal{T}^k$. Since $\gamma^k(V)$ is also the number of vertices of $g^k(V)$, we have:

$$3t^k = 2e^k - \gamma^k(V)$$

Moreover, from Euler's relation:

$$g^k = e^k - t^k + 1$$

It results from the two previous relations that, for every $k \in \{1, \ldots, n-1\}$,

$$e^k = 3g^k - \gamma^k(V) - 3 \tag{6.4}$$

$$t^k = 2g^k - \gamma^k(V) - 2 \tag{6.5}$$

For every subset T of $k+1 \geq 2$ points of V that admits a domain in $\mathcal{T}^k$, let e^k_T and t^k_T be the numbers of edges and of triangles of this domain. Since, from the domain convexity conjecture, the domain of T is either reduced to a line segment or is a triangulation of a convex polygon, we have, for every $k \in \{1, \ldots, n-1\}$,

$$e^k_T = 2t^k_T + 1 \tag{6.6}$$

Now, every edge and every domain triangle of $\mathcal{T}^k$ belongs to one, and only one, domain. Moreover, from Proposition 6.11, the number of domains of $\mathcal{T}^k$ is equal to the number of vertices of $\mathcal{T}^{k+1}$.

By summing relation (6.6) over the set of domains of $\mathcal{T}^k$ we then obtain, for every $k \in \{1, \ldots, n-1\}$,

$$e^k = 2t^k_{dom} + g^{k+1}$$

Hence, for every $k \in \{2, \ldots, n-1\}$,

$$e^k + e^{k-1} = 2(t^k_{dom} + t^{k-1}_{dom}) + g^{k+1} + g^k$$

Now, by construction and from Conjecture 2.25,

$$t^k_{dom} = t^{k-1}_{ter}$$

Since on the other hand,

$$t^{k-1} = t^{k-1}_{ter} + t^{k-1}_{dom}$$

it results that, for every $k \in \{2, \ldots, n-2\}$,

$$e^k + e^{k-1} = 2t^{k-1} + g^{k+1} + g^k \tag{6.7}$$

Using relations (6.4), (6.5), (6.7) we obtain then, for every $k \in \{2, \ldots, n-2\}$,

$$g^{k+1} - 2g^k + g^{k-1} = -\gamma^k(V) + \gamma^{k-1}(V) - 2$$

We can resolve this equation in the following way:

$$\begin{aligned}
g^{k+1} - 2g^k + g^{k-1} &= -\gamma^k(V) + \gamma^{k-1}(V) - 2 \\
g^k - 2g^{k-1} + g^{k-2} &= -\gamma^{k-1}(V) + \gamma^{k-2}(V) - 2 \\
g^{k-1} - 2g^{k-2} + g^{k-3} &= -\gamma^{k-2}(V) + \gamma^{k-3}(V) - 2 \\
&\vdots \\
g^4 - 2g^3 + g^2 &= -\gamma^3(V) + \gamma^2(V) - 2 \\
g^3 - 2g^2 + g^1 &= -\gamma^2(V) - \gamma^1(V) - 2
\end{aligned}$$

By multiplying the second equation by 2, the third by 3, ..., the last one by $k-1$, and by summing all of them we obtain:

$$g^{k+1} - kg^2 + (k-1)g^1 = -\gamma^k(V) - \gamma^{k-1}(V) \ldots - \gamma^2(V) + (k-1)\gamma^1(V) - k(k-1) \tag{6.9}$$

Now, the number of vertices of an order-1 triangulation is equal to the number of points of V, and, from Proposition 2.23, the number of vertices of an order-2 triangulation is equal to the number of edges of an order-1 triangulation. Thus, by using relation (6.4):

$$g^1 = n \tag{6.10}$$

$$g^2 = e^1 = 3n - \gamma^1(V) - 3 \tag{6.11}$$

By replacing these values in (6.9), we obtain, for every $k \in \{2, \ldots, n-1\}$,

$$g^{k+1} = 2(k+1)n - n - (k+1)^2 + 1 - \sum_{i=1}^{k} \gamma^i(V)$$

From (6.10) and (6.11) the relation still holds for $k = 0$ and $k = 1$, by setting $\sum_{i=1}^{0} = 0$. $\square$

Chapter 7

Conclusion

The first result in this dissertation is the finding of a new invariant of the number of k-sets of a set of points V in the plane. To obtain this result, we have introduced the notion of convex inclusion chain of V, that is an ordering of the points of V such that every point is outside the convex hull of the points that precede it. We have shown that the total number of k-sets of all the initial subsequences of a convex inclusion chain of V is an invariant of V, that is, it does not depend on the choice of the convex inclusion chain. Moreover, this number of k-sets is equal to the number of order-k Voronoi regions of V.

This result leads to compute the size of an order-k Voronoi diagram using a completely different method than the usual one. We can hope then, that the study of convex inclusion chains in a higher dimensions (> 2) can help us resolving the open problem of the size of order-k Voronoi diagrams in higher dimensions.

The previous result on the number of k-sets was obtained using the k-set polygon of the considered point set. This k-set polygon is the convex hull of the centroids of all k-point subsets of V. Moreover, we have extended two classical convex hull construction algorithms to the construction of the k-set polygon: An incremental algorithm and a divide and conquer algorithm.

The incremental algorithm was studied in the case where the points were added in the order they appear in a convex inclusion chain; that is, the new added point is separable by a straight line from the previously inserted points. Moreover, this algorithm was efficiently implemented when the convex inclusion chain formed a simple polygonal line.

In the divide and conquer algorithm, the set of points was divided into two left and right subsets, not necessarily disjoint, but such that the points that do not belong to one of the subsets are separable from the others by a vertical straight line.

The straight line separability constraint imposed in both algorithms put us in the situation where the edges to remove from a k-set polygon form a polygonal line. This property is not verified in the general case.

Now, it would be interesting to extend other classical convex hull construction algorithms to the construction of the k-set polygon; in particular, the Quick Hull algorithm known to be the most efficient algorithm in practice. However, the general case for most of these algorithms must be studied, that is, the case where a point appears inside the convex hull of the already processed points.

In the complexity study of our divide and conquer algorithm, a supplementary $\log(n/k)$ factor appeared in comparison to the best known k-set finder algorithm. This factor comes from an overestimation while analyzing the complexity. We think that this factor can be removed by a refined analysis of the number of edges created by the algorithm.

In the last chapter, we tried to understand why the number of k-sets of a convex inclusion chain of a set of points in the plane is equal to the number of order-k Voronoi regions of the same point set. To this aim, we used the dual of the order-k Voronoi diagram, called order-k Delaunay triangulation, whose vertices are the centroids of the subsets of k points that define the order-k Voronoi regions. This triangulation belongs to a family of triangulations called order-k centroid triangulations. An order-k centroid triangulation is defined in a constructive way from an order-$(k-1)$ centroid triangulation (where the order-1 triangulation is an arbitrary triangulation of the considered point set).

We have shown that the centroids of the k-sets of a convex inclusion chain of a set V of points in the plane are the vertices of an order-k centroid triangulation of V, thereby establishing a link between the k-sets of a convex inclusion chain and the order-k Voronoi regions. To complete the argument, it remains to prove that all order-k centroid triangulations have the same number of vertices. The problem is that, for $k > 3$, we do not know which are the other triangulations that belong to the family of centroid triangulations. In fact, there is no proof showing that the method that builds an order-k triangulation from an order-$(k-1)$ triangulation works for $k > 3$, for triangulations other than the order-k Delaunay and the triangulations obtained from the convex inclusion chains as defined in this dissertation.

However, we were successful in giving a sufficient condition that needs to be verified by all centroid triangulations so they all have the same number of vertices: All their triangles that are defined using the same $k+1$ points must form a convex set.

The difficulty in proving the existence of other centroid triangulations comes from the recursive definition of these triangulations. Thereby, the existence of an order-k centroid triangulation depends on the existence of a sequence of order-i centroid triangulations, for every $i < k$. We deem it necessary to find a direct geometrical characterization of an order-k centroid triangulation that does not depend on the lower order triangulations. The practical experimentations that we have made seem to indicate that every vertex of an order-k centroid triangulation of V is the centroid of a subset of k points of V that is separable from the rest of the points of V by a convex curve.

Finally, we have proved that by pre-sorting the set of points, it is possible to construct a particular order-k centroid triangulation without constructing the lower orders. Thus, we have obtained a more efficient algorithm than the one deduced from the definition of the centroid triangulations.

The algorithmic question that arises now, is to know if it is possible to transform an order-k centroid triangulation into an order-k Delaunay triangulation by a sequence of local modifications (by only generating intermediate triangulations that are centroid triangulations as well). In the case of classical triangulations such an algorithm is called a flip algorithm. The reason that allows us to think that such an algorithm can exist is that, as in the case of classical triangulations, the order-k Delaunay triangulation is the projection of a convex surface of dimension 3. More precisely, the order-k Delaunay triangulation of V is the projection in the plane of the lower part of the k-set polytope of the points of V lifted on a 3-dimensional paraboloid. It results that the lifting of an order-k centroid triangulation that is not Delaunay is above the lifting of the order-k Delaunay triangulation. For a flip algorithm to work as in the case of classical triangulations, it suffices then to perform local improvements that bring the lifting down, thereby becoming closer to the order-k Delaunay triangulation.

Bibliography

[AF99] A. Andrzejak and K. Fukuda. Optimization over k-set polytopes and efficient k-set enumeration. In *Proc. 6th Workshop Algorithms Data Struct.*, volume 1663 of *Lecture Notes Comput. Sci.*, pages 1–12. Springer-Verlag, 1999.

[AGSS89] A. Aggarwal, L. J. Guibas, J. B. Saxe, and P. W. Shor. A linear-time algorithm for computing the voronoi diagram of a convex polygon. *Discrete & Computational Geometry*, 4:591–604, 1989.

[AS92] F. Aurenhammer and O. Schwarzkopf. A simple on-line randomized incremental algorithm for computing higher order voronoi diagrams. *Int. J. Comput. Geometry Appl.*, 2(4):363–381, 1992.

[Aur91] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

[BJ02] G. S. Brodal and R. Jacob. Dynamic planar convex hull. In *Proc. 43rd IEEE Sympos. Found. Comput. Sci*, pages 617–626, 2002.

[CGL85] B. Chazelle, L. J. Guibas, and D. T. Lee. The power of geometric duality. *BIT*, 25(1):76–90, 1985.

[CK70] D. R. Chand and S. S. Kapur. An algorithm for convex polytopes. *J. ACM*, 17(1):78–86, 1970.

[CP86] B. Chazelle and F. P. Preparata. Halfspace range search: An algorithmic application of k-sets. *Discrete & Computational Geometry*, 1:83–93, 1986.

[CSY87] R. Cole, Micha Sharir, and C. K. Yap. On k-hulls and related problems. *SIAM J. Comput.*, 16:61–77, 1987.

[Dey98] T. K. Dey. Improved bounds on planar k-sets and related problems. *Discrete Comput. Geom.*, 19:373–382, 1998.

[Ede87] H. Edelsbrunner. *Algorithms in combinatorial geometry.* Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[EHSS89] H. Edelsbrunner, H. Hasan, N. Seidel, and Shen. Circles through two points that always enclose many points. *Geometriae Dedicata.*, 32:1–12, 1989.

[ELSS73] P. Erdös, L. Lovász, A. Simmons, and E. G. Straus. Dissection graphs of planar point sets. *A Survey of Combinatorial Theory (Symps. Colorado State Univ.)*, pages 139–149, 1973.

[ERvK93] H. Everett, J.-M. Robert, and M. J. van Kreveld. An Optimal Algorithm for the (<= k)-Levels, with Applications to Separation and Transversal Problems. In *Symposium on Computational Geometry*, pages 38–46, 1993.

[EVW97] H. Edelsbrunner, P. Valtr, and E. Welzl. Cutting dense point sets in half. *Discrete Comput. Geom.*, 17:243–255, 1997.

[Jar73] R. A. Jarvis. On the identification of the convex hull of of a finite set of points in the plane. *Inf. Process. Lett.*, 2:18–21, 1973.

[Kal84] M. Kallay. The complexity of incremental convex hull algorithms in $\mathbb{R}^d$. *Inf. Process. Lett.*, 19(4):197, 1984.

[Lee82] D. T. Lee. On k-nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Comput.*, C-31:478–487, 1982.

[Lov71] L. Lovász. On the number of halving lines. *Ann. Univ. Sci. Budapest, Etvs, Sect. Math.*, 14:107–108, 1971.

[LS07] Y. Liu and J. Snoeyink. Quadratic and cubic b-splines by generalizing higher-order voronoi diagrams. In J. Erickson, editor, *Symposium on Computational Geometry*, pages 150–157. ACM, 2007.

[Mel87] A. Melkman. On-line construction of the convex hull of a simple polyline. *Inform. Process. Lett.*, 25:11–12, 1987.

[Nea04] M. Neamtu. Delaunay configurations and multivariate spline: A generalization of a result of B. N. Delaunay. *Trans. Amer. Soc.*, 2004.

[OS99] S. Onn and B. Sturmfels. Cutting corners. *Advances in Applied Mathematics*, 23:29–48, 1999.

[Ove83] M. H. Overmars. *The Design of Dynamic Data Structures*, volume 156 of *Lecture Notes Comput. Sci.* Springer-Verlag, Heidelberg, West Germany, 1983.

[OvL81] M. H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *J. Comput. Syst. Sci.*, 23:166–204, 1981.

[Pec85] G. W. Peck. On k-sets in the plane. *Discrete Math.*, 56:73–74, 1985.

[PH77] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM*, 20(2):87–93, 1977.

[PS85] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, NY, 1985.

[Sch95] D. Schmitt. *Sur les diagrammes de Delaunay et de Voronoï d'ordre k dans le plan et dans l'espace.* Ph.D. thesis, Université de Haute–Alsace, Mulhouse, France, 1995.

[SS91] D. Schmitt and J.-C. Spehner. On Delaunay and Voronoi diagrams of order k in the plane. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 29–32, 1991.

[SS98] D. Schmitt and J.-C. Spehner. Order-k Voronoi Diagrams, k-Sections, and k-sets. In *JCDCG*, pages 290–304, 1998.

[SS06] D. Schmitt and J.-C. Spehner. k-set Polytopes and Order-k Delaunay Diagrams. In *ISVD*, pages 173–185. IEEE Computer Society, 2006.

[Tót01] G. Tóth. Point sets with many k-sets. *Discrete Comput. Geom.*, 26(2):187–194, 2001.

[YKII88] P. Yamamoto, K. Kato, K. Imai, and H. Imai. Algorithms for vertical and orthogonal l1 linear approximation of points. In *Symposium on Computational Geometry*, pages 352–361, 1988.