Werk

Titel: A New Duality Result Concerning Voronoi Diagrams.

Autor: Aurenhammer, F.

Jahr: 1990

PURL: https://resolver.sub.uni-goettingen.de/purl?362609810_0005|log19

Kontakt/Contact

Digizeitschriften e.V.
SUB Göttingen
Platz der Göttinger Sieben 1
37073 Göttingen

info@digizeitschriften.de

Discrete Comput Geom 5:243-254 (1990)

Discrete & Computational Geometry
© 1990 Springer-Verlag New York Inc.

A New Duality Result Concerning Voronoi Diagrams*

Franz Aurenhammer

Institutes for Information Processing, Technical University of Graz and Austrian Computer Society, Schiesstattgasse 4a, A-8010 Graz, Austria

Abstract. A new duality between order-k Voronoi diagrams in E^d and convex hulls in E^{d+1} is established. It implies a reasonably simple algorithm for computing the order-k diagram for n points in the plane in $O(k^2 n \log n)$ time and optimal $O(k(n-k))$ space.

1. Introduction

Given a set P of n points in Euclidean d-space E^d, $d \geq 1$, and an integer k, $1 \leq k \leq n-1$, the *order-k Voronoi diagram*, $k - V(P)$, of P subdivides E^d into maximal regions (called *cells*), such that any point within a fixed cell has the same k nearest points in P. More formally, $k - V(P)$ contains, for each *k-subset* S of P (i.e., $S \subsetneq P$ and $|S| = k$), the cell

$$\text{cell}(S) = \{x \in E^d \mid \delta(x, s) \leq \delta(x, p), \forall s \in S, \forall p \in P - S\},$$

where δ denotes the Euclidean distance function. Since $\{x \in E^d \mid \delta(x, s) = \delta(x, p)\}$ is the symmetry-hyperplane of s and p, cell(S) is the intersection of $k(n-k)$ closed halfspaces of E^d and thus a (possibly degenerate) closed and convex polyhedron in E^d. The interiors of two distinct cells, by definition, do not intersect, and the cells of $k - V(P)$ cover E^d; see Fig. 1 for an illustration. For $0 \leq j \leq d-1$, the j-dimensional polyhedra bounding the cells are called *j-faces* of $k - V(P)$ or, synonymously, *facets* for $j = d-1$, *edges* for $j = 1$, and *vertices* for $j = 0$.

Voronoi diagrams have been independently discovered and used in various areas of science. Once introduced in computer science by Shamos and Hoey [11], they have received a considerable deal of attention in computational geometry within the last decade. The construction of $1 - V(P)$ in E^d was optimally solved

---

* Research was supported by the Austrian Fond zur Foerderung der wissenschaftlichen Forschung.

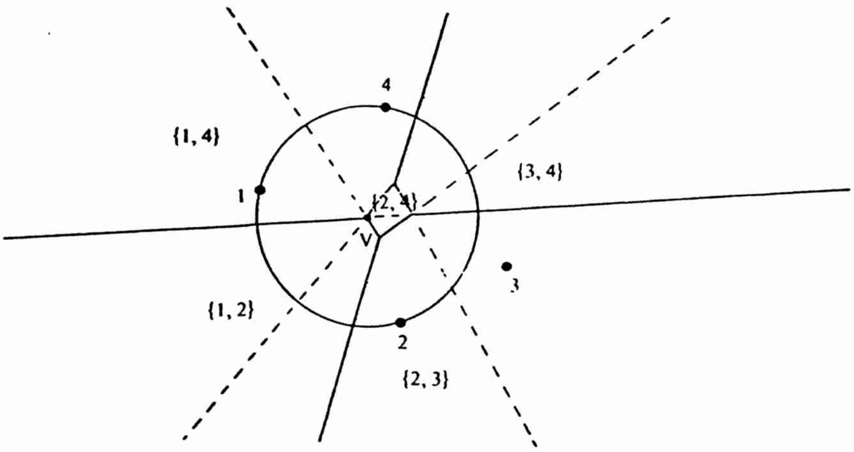

Fig. 1. $1-V(P)$ (dashed line) and $2-V(P)$ (solid line) for $P=\{1,2,3,4\}$. The cells of $2-V(P)$ have their sets associated.

in [11] for $d=1,2$ and in [3] and [9] for $d\geq 3$, d odd. The latter approach is based on a duality between $1-V(P)$s and convex hulls in E^{d+1} and, by a recent result in [10], is optimal to within a time factor of $\log n$ for $d>3$, d even. While [6] succeeded in devising an algorithm which computes all $k-V(P)$s in E^d, for $k=1,\ldots,n-1$, in optimal time and space $O(n^{d+1})$ exploiting a geometric correspondence to arrangements of hyperplanes in E^{d+1}, several attempts have been made to compute the particular diagram $k-V(P)$ efficiently. In E^2 the first method is due to Lee [7] who also showed that the number of cells, edges, and vertices is bounded by $O(k(n-k))$ in this case. The method iteratively derives $k-V(P)$ from $(k-1)-V(P)$ and takes $O(k^2 n\log n)$ time and $O(k^2(n-k))$ storage. Using the geometric background in [6], this result was improved to $O(k(n-k)\sqrt{n}\log n)$ time and $O(k(n-k))$ space [5] and recently to $O(n^2\log n+k(n-k)\log^2 n)$ time, $O(k(n-k))$ space or $O(n^2+k(n-k)\log^2 n)$ time, $O(n^2)$ space, respectively [4]. Nevertheless, Lee's method is most time-efficient for $k<\sqrt{n/\log n}$.

The main contribution of this paper is a tailor-made algorithm for constructing low-order Voronoi diagrams in E^2. It resembles Lee's algorithm in that it works iteratively and results in the same time complexity, but differs in that it is conceptually simpler, has optimal space requirement, and generalizes to higher dimensions. The new construction method (outlined in Section 3) relies on a duality between $k-V(P)$s in E^d and convex hulls in E^{d+1} that is described in Section 2 and is of interest in its own right. Section 4 offers a discussion of the results and mentions some extensions.

2. The Duality to Convex Hulls

Let C and C' be two convex polyhedra in E^{d+1} and let the j-dimensional polyhedra bounding them be their *j-faces*, for $0\leq j\leq d$. C and C' are said to be *dual* if there is a one-to-one correspondence Ψ between the j-faces of C and the $(d-j)$-faces of C' such that $f \subsetneq g$, for any two faces f and g of C, iff $\Psi(f) \supsetneq \Psi(g)$. Various polyhedral partitions of E^d can be viewed as the affine degeneracies of the boundaries of convex polyhedra in E^{d+1}, see, e.g., [2]. Thus the notion of duality remains meaningful if C is replaced by such a partition. This section demonstrates that, given a finite point-set P in E^d, it is possible to construct a point-set Q_k in E^{d+1} such that the diagram $k-V(P)$ is dual to certain parts of the convex hull of Q_k, for $1 \le k \le |P|-1$. This was known for the special case $k=1$, see, e.g., [3]. Our result is a direct, but by no means trivial, generalization.

We start by clarifying what is meant by "certain parts" of the convex hull, conv Q, of a finite point-set Q in E^{d+1}. Conv Q is the convex polyhedron that represents the intersection of all closed halfspaces of E^{d+1} containing Q. The *lower part*, low Q, of conv Q is the collection of all faces f of conv Q such that there exists a hyperplane h with $f \subseteq h$, h *nonvertical* (i.e., not parallel to the x_{d+1}-axis of E^{d+1}), and conv Q *above* h (i.e., conv Q lies in the closed halfspace that is bounded by h and that contains the point on the x_{d+1}-axis at $+\infty$).

Let a set P of n points in E^d be given. For each k, $1 \le k \le n-1$, P can be associated with a *dual point-set* Q_k in E^{d+1} as follows:

(i) We identify E^d with the hyperplane h_0: $x_{d+1}=0$ in E^{d+1}.
(ii) We define $Q_k = \{q(S) = (\xi_1, \ldots, \xi_{d+1}) | S$ is a k-subset of $P\}$, where $(\xi_1, \ldots, \xi_d) = \sum_{p \in S} p$ and $\xi_{d+1} = \sum_{p \in S} p^2$.

Q_k has the following obvious property. For two k-subsets S and S' of P with $S' = (S \cup \{p_j\}) - \{p_i\}$ and $q(S') = (\eta_1, \ldots, \eta_{d+1})$,

$$(\eta_1, \ldots, \eta_d) = (\xi_1, \ldots, \xi_d) + p_j - p_i \quad \text{and} \quad \eta_{d+1} = \xi_{d+1} + p_j^2 - p_i^2 \text{ holds.}$$

By means of the terminology introduced, the main assertion of this section can be stated as below.

Theorem 1. *$k-V(P)$ is dual to* low Q_k.

For explanatory reasons we shall assume that P is in *general position*, i.e., $|P| \ge d+1$ and no $d+1$ $(d+2)$ points in P lie on a common hyperplane (sphere) in h_0. In this nondegenerate case, conv Q_k contains no vertical facet and each of its j-faces is a *j-simplex* (contains exactly $j+1$ vertices). Moreover, the following auxiliary lemma holds, the proof of which is left to the interested reader.

Lemma 1.

(i) *$v \in h_0$ is a vertex of $k-V(P)$ iff there is a sphere σ centered at v which partitions P into P_-, P_0, and P_+ such that*

$$P_0 \subsetneq \sigma \quad \textit{and} \quad |P_0| = d+1,$$

$$P_- = \left(\bigcup_{v \in \text{cell}(S)} S \right) - P_0 \subsetneq \text{int}\, \sigma,$$

$$P_+ \subsetneq \text{ext}\, \sigma,$$

where $\operatorname{int}\sigma$ *and* $\operatorname{ext}\sigma$ *denote the open ball bounded by* σ *and its open complement, respectively.*

(ii) *For any two distinct k-subsets S and S' of P with* $v \in \operatorname{cell}(S) \cap \operatorname{cell}(S')$ *we have* $S' = (S \cup \{p_j\}) - \{p_i\}$ *for some distinct* $p_j, p_i \in P_0$.

Proof of Theorem 1. Observe first that $k - V(P)$ and low Q_k are dual iff, for each vertex v of $k - V(P)$, $f = \operatorname{conv}\{q(S) | v \in \operatorname{cell}(S)\}$ is a d-face of low Q_k. By definition of low Q_k, f is a d-face of low Q_k iff $q(S')$ lies above the hyperplane h containing f for all k-subsets S' of P with $v \notin \operatorname{cell}(S')$, or by Lemma 1 equivalently, iff $q(S_+)$ lies above h for all $S_+ = (S \cup \{p_+\}) - \{p\}$ with $p \in P_0$ and $p_+ \in P_+$.

Without loss of generality, let v be such that $q(S)$ coincides with the origin. Then the vertical projection of $q(S_+)$ onto h_0 is given by $p_+ - p$, and its x_{d+1}-coordinate by $p_+^2 - p^2$. Lemma 1 implies $\delta(p_+, v) > \delta(p, v)$, that is, $p_+^2 - p^2 > 2v(p_+ - p)$ holds. In conjunction with the fact below, this is equivalent to $q(S_+)$ lying above h, and thus implies the theorem. □

Fact. $2v(p_+ - p)$ *is the* x_{d+1}*-coordinate of the vertical projection of* $q(S_+)$ *onto h.*

Proof. By Lemma 1(ii) and the definition of h, the assertion is true iff there is some $\lambda \in E^d$ such that

$$(\cdots (p_j - p_i) \cdots)\lambda = p_+ - p \quad \text{and} \quad \begin{pmatrix} \vdots \\ p_j^2 - p_i^2 \\ \vdots \end{pmatrix} \lambda = 2v(p_+ - p),$$

for d pairs (p_j, p_i) with $p_j, p_i \in P_0$ and $p_j \neq p_i$. Elimination of $(p_+ - p)$ and λ yields d identities $2v(p_j - p_i) = p_j^2 - p_i^2$.

In geometric terms this means that v is the intersection of the d symmetry-hyperplanes defined in h_0 by (p_j, p_i), that is, P_0 lies on a sphere σ centered at v. Since v was a vertex of $k - V(P)$, Lemma 1(i) proves the fact. □

We only mention that the proof of Theorem 1 can be extended to the case where no restrictions on P are drawn. However, in the sequel we adopt the convention that P is in a general position and refer to [5] for algorithmic methods which remove degeneracies.

Figure 2 illustrates how conv Q_2 is obtained from the point-set P shown in Fig. 1. The reader may examine the duality between the j-faces of low Q_2 and the $(2-j)$-faces of $2 - V(P)$.

3. An Iterative Construction Method

This section puts the geometric investigations of Section 2 to use by developing a new algorithm for constructing order-k Voronoi diagrams.

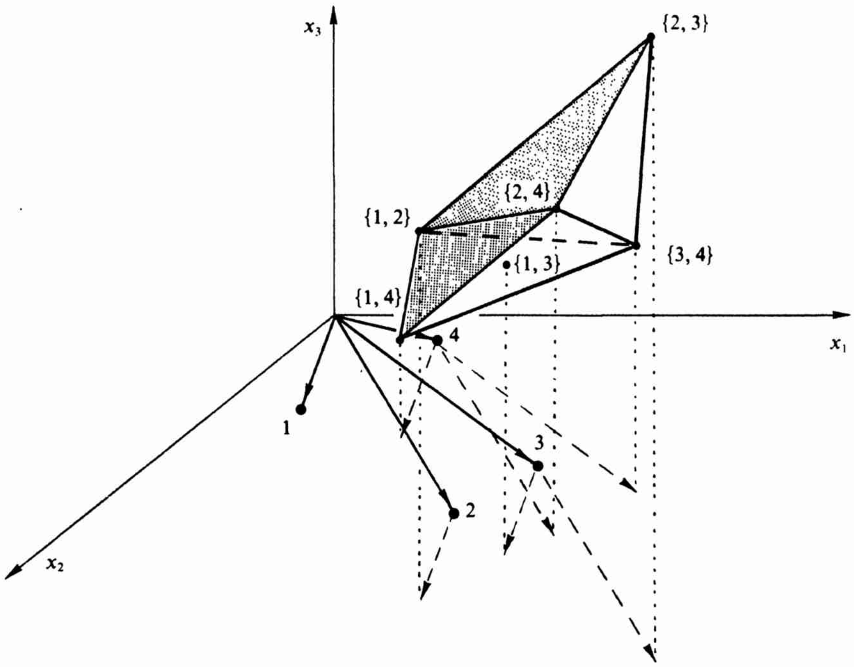

Fig. 2. Relationship between P and conv Q_2.

3.1. *The Overall Structure*

According to Theorem 1, the construction of $k-V(P)$ in E^d is essentially equivalent to determining the lower part of a convex hull conv Q_k in E^{d+1}. Since Q_k can be calculated directly from P and algorithms for determining the convex hull of a finite point-set in E^{d+1} are well established, this implies a direct method for computing $k-V(P)$ in arbitrary dimensions. However, P realizes $\binom{n}{k}$ k-subsets for $|P|=n$, not each of which may give rise to a nonempty cell in $k-V(P)$. In the dual environment this means that only a few of the $\binom{n}{k}$ points of Q_k may define vertices of low Q_k so that, in general, a very inefficient algorithm would result.

To remedy this shortcoming, we apply a strategy (also used for the case $d=2$ and in different terms in [7]) which exploits the information inherent in low Q_{k-1} to compute exactly those points of Q_k that lie on low Q_k. It relies on the following lemma.

Lemma 2. *For an edge e of* low Q_{k-1}, *let* pair$(e)=(p_j, p_i)$ *if e is considered as directed from $q(S_i)$ to $q(S_j)$ and if $S_j=(S_i\cup\{p_j\})-\{p_i\}$. Furthermore, let* set$(e)=S_i\cup S_j$.

For $i=1, 2, 3$, *the edges* e_i *bound a 2-face (triangle) t of* low Q_{k-1} *iff* $\tau(t) = \text{conv}\{q(S) | S = \text{set}(e_i)\}$ *either is a vertex or a triangle of* low Q_k. *Moreover,* $\text{pair}(\varepsilon_i) = \text{pair}(e_i)$ *holds for the edges* ε_i *of* $\tau(t)$.

Proof. t is dual to a $(d-2)$-face $g = \text{cell}(S_1) \cap \text{cell}(S_2) \cap \text{cell}(S_3)$ of $(k-1)-V(P)$. Now the following two situations may occur.

(i) g is no $(d-2)$-face of $k-V(P)$ iff $S_1 = S' \cup \{p_2, p_3\}$, $S_2 = S' \cup \{p_1, p_3\}$, $S_3 = S' \cup \{p_1, p_2\}$ holds for some $(k-3)$-subset $S' \subsetneq P$ and distinct $p_1, p_2, p_3 \in P - S'$. (The interested reader may verify this equivalence using Lemma 1.) Thus $\text{set}(e_i) = S' \cup \{p_1, p_2, p_3\} = S$, for $i = 1, 2, 3$, such that $\tau(t) = q(S)$. Observe that $\text{pair}(e_1) = (p_2, p_3)$, $\text{pair}(e_2) = (p_3, p_1)$, and $\text{pair}(e_3) = (p_1, p_2)$ holds if the e_i's are directed as shown in Fig. 3(a).

(ii) g is a $(d-2)$-face of $k-V(P)$ iff we have $S_1 = S'' \cup \{p_1\}$, $S_2 = S'' \cup \{p_2\}$, and $S_3 = S'' \cup \{p_3\}$ for some $(k-2)$-subset $S'' \subsetneq P$ and distinct $p_1, p_2, p_3 \in P - S''$. (Again the easy proof is left to the reader.) Moreover, $g = \bigcap_{i=1,2,3} \text{cell}(\text{set}(e_i))$ and $\text{set}(e_i) = (S'' \cup \{p_1, p_2, p_3\}) - \{p_i\}$ holds. This means that $\tau(t)$ is a triangle of low Q_k with vertices $q_i = q(\text{set}(e_i))$, for $i = 1, 2, 3$. In addition, if e_i is directed from $q(S_j)$ to $q(S_r)$, we have $\text{pair}(e_i) = (p_r, p_j) = \text{pair}(\varepsilon_i)$ for ε_i directed from q_j to q_r $(1 \le i, j, r \le 3)$; see Fig. 3(b). □

The overall structure of the iterative algorithm that constructs $k-V(P)$ can now be described:

Step 1. Compute $V_1 = Q_1$ from P and construct conv V_1.

Step 2. For m running from 2 to k, derive the set $V_m \subseteq Q_m$ of vertices of low Q_m from the triangles of low V_{m-1}, and construct conv V_m.

Step 3. Dualize low V_k to $k-V(P)$.

The difference between Lee's [7] approach and ours should be observed. To obtain $m-V(P)$ from $(m-1)-V(P)$ he constructs a particular order-1 diagram

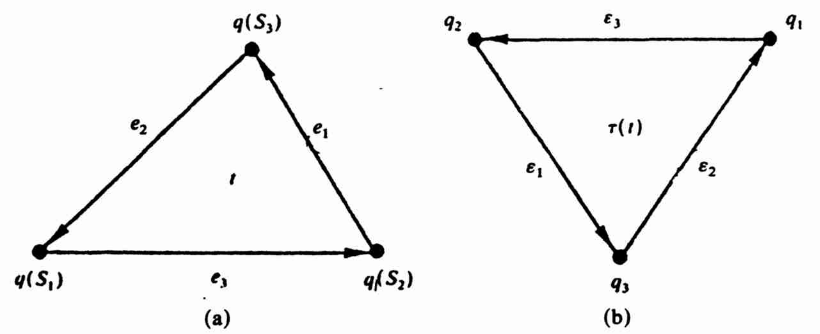

Fig. 3. (a) Triangle t of low V_{m-1} and (b), if existent, its corresponding triangle $\tau(t)$ of low V_m.

for each cell of $(m-1)-V(P)$. In the dual environment, this would mean constructing (the lower part of) a particular convex hull for each vertex of low V_{m-1} and then putting these parts together to obtain low V_m. Step 2 above obtains low V_m directly by constructing only one convex hull, conv V_m, thus simplifying the main part of the algorithm.

3.2. Detailing the Steps

Before detailing and formalizing the particular steps taken by the algorithm, a suitable scheme for storing conv V_m, i.e., a convex polyhedron C in E^{d+1}, has to be specified. Two faces f and g of C are said to be *incident* iff their dimensions differ by one and either $f \subsetneq g$ or $g \subsetneq f$. For technical reasons, C and the empty set $\varnothing$ are considered as a $(d+1)$-face and a (-1)-face, respectively, so that C is incident upon its facets and $\varnothing$ is incident upon the vertices of C. Now the combinatorial structure of C is reflected by the *incidence lattice* $I(C)$ of C. This involves storing a node for each j-face of C, for $-1 \le j \le d+1$, and associating nodes of incident faces of C via pointers. Storing the coordinates of C's vertices determines the position of C in E^{d+1}. The representation of C by $I(C)$ is appropriate for our purposes since $I(C)$ can be computed from the output produced by the convex-hull algorithms in [8]-[10] in time proportional to the number of faces of C. Note that $I(C)$ allows us to determine for each j-face f of C the n' $(j-1)$-faces and the n'' $(j+1)$-faces of C incident upon f in $O(n'+n'')$ time.

According to Lemma 2 and the definition of $q(S)$, Step 2 of the previously sketched algorithm requires the m-subsets set(e) for the edges e of low V_{m-1} for the computation of the vertices $q(\text{set}(e)) \in V_m$. Space can be saved by employing the following implicit storing scheme rather than storing set(e) for each e explicitly.

For one vertex v_0 of low V_{m-1}, its corresponding $(m-1)$-subset S_0 is stored explicitly.

Each edge e of low V_{m-1}, incident upon vertices $q(S)$ and $q(S')$ say, is considered as directed from $q(S)$ to $q(S')$. Together with e's direction we store the ordered pair pair$(e)=(p',p)$ which is uniquely defined by $S'=(S\cup\{p'\})-\{p\}$, $p'\neq p$, according to Lemma 1(ii).

The algorithm now visits each triangle t of low V_{m-1}. If the sets set(e_i) are not identical for the edges e_i incident upon t then the vertices $q(\text{set}(e_i))$ of the triangle $\tau(t)$ of low V_m are computed and the edges ε_i of $\tau(t)$ get assigned pair(ε_i) as indicated in Lemma 2.

After the construction of conv V_m, we are left with the problem of associating the remaining edges ε' of low V_m with pair(ε'). To this end, let us study the similarities in the facial structure of $(m-1)-V(P)$ and $m-V(P)$. Observe that a cell cell(S) of $(m-1)-V(P)$ in E^d splits into parts of cells cell$(S_1), \ldots,$ cell(S_r)

of $m-V(P)$. For each facet f incident upon cell(S_i) $(1\le i\le r)$ whose relative interior intersects cell(S) there is a $(d-2)$-face g of cell(S) with $g\subsetneq f$. In the dual environment this is equivalent to vertex $q(S)$ of low V_{m-1} mapping to vertices $q(S_1),\ldots,q(S_r)$ of low V_m, and edges ε dual to facets f being incident upon vertices $q(S_i)$ $(1\le i\le r)$ and defining the relative boundary $\Delta(S)$ of a simplicial surface $\Sigma(S)$ on low V_m. By Lemma 2, pair(ε) for each ε in $\Delta(S)$ has already been computed since the $(d-2)$-faces $g\subsetneq$ cell$(S)\cap f$ occur in both diagrams $(m-1)-V(P)$ and $m-V(P)$.

To compute pair(ε') for each edge ε' in $\Sigma'(S)=\Sigma(S)-\Delta(S)$ we observe that, if ε' exists, it is incident upon two vertices $q(S_i)$, $q(S_j)$ in $\Delta(S)$ $(1\le i<j\le r)$, since $f'=$ cell$(S_i)\cap$ cell(S_j) is its dual facet. (Note that $f'\subsetneq$ cell(S) is equivalent to ε' in $\Sigma'(S)$.) This implies that no vertex of $\Sigma(S)$ is in $\Sigma'(S)$, such that $\Sigma(S)$ always contains some triangle incident upon edges ε_1, ε_2 in $\Delta(S)$ and edge ε' in $\Sigma'(S)$. Therefore we can use the following.

Observation. Let pair$(\varepsilon_1)=(p_1,p)$ and pair$(\varepsilon_2)=(p,p_2)$. Then pair$(\varepsilon')=(p_1,p_2)$ holds if ε_1, ε_2, ε' are considered as directed edges with $\varepsilon'=\varepsilon_1+\varepsilon_2$.

Splitting off $\Sigma(S)$ all simplices for which ε' was the only edge whose pair was not computed yields again a simplicial surface whose boundary edges have their pairs computed. This finally shows that all remaining edges ε' of low V_m can be assigned pair(ε') by means of the observation above.

3.3. *The Algorithm*

Our investigations so far result in the more formal description below of the algorithm that computes the order-k Voronoi diagram $k-V(P)$ of a finite set P of points in E^d, for $d\ge 1$. To aid the intuition, an edge e of low V_{m-1} is referred to as a *green* (*white*, *red*) edge if pair(e) and $q($set$(e))$ (only pair(e), neither pair(e) nor $q($set$(e))$) have been computed.

Algorithm. CONSTRUCT DIAGRAM

Step 1. Compute V_1 and construct low V_1.

1.1. Assign $V_1=\{q(\{p\})|p\in P\}$, taking $q(\{p\})=(\xi_1,\ldots,\xi_{d+1})$, for $(\xi_1,\ldots,\xi_d)=p$ and $\xi_{d+1}=p^2$.
1.2. Construct conv V_1 in E^{d+1}, using the algorithms in [8]-[10] for $d=1$, 2, $d\ge 3$ and odd, $d>3$ and even, respectively, such that conv V_1 is stored in $I($conv $V_1)$.
1.3. For each edge e of low V_1, set pair$(e)=(p',p)$ if e is assigned the direction from $q(\{p\})$ to $q(\{p'\})$, and color e white. e belongs to low V_1 iff e is contained in a facet of conv V_1 whose hyperplane bounds conv V_1 below. For a particular vertex $v_0=q(\{p\})$ of low V_1, set $S_0=\{p\}$ for its 1-subset S_0. In addition, let ST be an initially empty stack.

Step 2. For m running from 2 to k do: Construct low V_m.

2.1. Compute V_m. Initially, $V_m = \emptyset$. Observe that each edge e of low V_{m-1} is white. Let e_0 be an edge incident upon (and directed toward) v_0 and let t_0 be a triangle incident upon e_0. For pair$(e_0) = (p', p)$ we have set$(e_0) = S_0 \cup \{p\}$. Assign $S_0 = \text{set}(e_0)$, $q(S_0) = (\xi_1, \ldots, \xi_d, 0)$ with $(\xi_1, \ldots, \xi_d) = v_0 + p$, $v_0 = q(S_0)$, and $V_m = V_m \cup \{v_0\}$. Color e_0 green and push t_0 onto *ST*.

While *ST* is nonempty do: Remove the first triangle t from *ST*. If necessary, reverse the order of the pairs of t's edges e_1, e_2, e_3 such that they are directed as shown in Fig. 3(a). At least one edge, e_1 say, is green so that $q(\text{set}(e_1)) = (\xi_1, \ldots, \xi_{d+1})$ has been computed.

For each white edge e_i of t do: If t's pairs are of the form as in part (i) of the proof of Lemma 2 then assign $q(\text{set}(e_i)) = q(\text{set}(e_1))$. Else we have pair$(e_1) = (p_3, p_2)$, pair$(e_2) = (p_1, p_3)$, and pair$(e_3) = (p_2, p_1)$. Compute $q(\text{set}(e_i)) = (\eta_1, \ldots, \eta_{d+1})$ using $(\eta_1, \ldots, \eta_d) = (\xi_1, \ldots, \xi_d) + p_1 - p_r$ and $\eta_{d+1} = \xi_{d+1} + p_1^2 - p_r^2$ (for $r = 3$ if $i = 2$ and $r = 2$, otherwise). Color e_i green and set $V_m = V_m \cup \{q(\text{set}(e_i))\}$. For the edges ε_i of low V_m directed from $q(\text{set}(e_j))$ to $q(\text{set}(e_r))$ do pair$(\varepsilon_i) = (p_r, p_j)$ for $1 \le i, j, r \le 3$.

Push all triangles of low V_{m-1} which are incident upon e_1, e_2, or e_3 and some white edge onto *ST*.

2.2. Construct conv V_m (see Step 1.2).

2.3. Calculate pair(ε) for each edge ε of low V_m. For each ε do: If pair(ε) has been assigned to ε in Step 2.1 then color ε white else color ε red. Push all triangles of low V_m which are incident upon exactly one red edge onto *ST*.

While *ST* is nonempty do: Remove the first triangle τ from *ST*. τ is incident upon two white edges ε_1, ε_2 and one red edge ε_3. If necessary, change the pairs of the ε_i's such that their associated directions imply $\varepsilon_3 = \varepsilon_1 + \varepsilon_2$. For pair$(\varepsilon_1) = (p_1, p)$ and pair$(\varepsilon_2) = (p, p_2)$, set pair$(\varepsilon_3) = (p_1, p_2)$ and color ε_3 white. Push all triangles incident upon ε_3 and upon exactly one red edge onto *ST*.

Step 3. Dualize low V_k to $k - V(P)$ in E^d.

3.1. Replace each j-face in $I(\text{low } V_k)$ by a $(d-j)$-face, for $j = -1, \ldots, d+1$, which yields $I(k - V(P))$.

3.2. For each vertex v of $k - V(P)$ which arises from a d-face f of low V_k do: Compute the union P_0 of all pair(e) over all edges e of f that are incident upon a fixed vertex. (Observe that $|P_0| = d+1$.) v gets assigned the coordinates of the center of the unique sphere σ in E^d with $P_0 \subsetneq \sigma$. This completes the construction of $k - V(P)$.

Theorem 2. *Let* size(d, k) *and* reg(d, k) *denote the maximal number of faces and of cells of* $k - V(P)$ *in* E^d, *respectively, and let* $T_d(r)$ *be the time needed to construct the convex hull of r points in* E^d. *Algorithm CONSTRUCT DIAGRAM requires*

$$O\left(\sum_{m=1}^{k} T_{d+1}(\text{reg}(d, m))\right) \textit{ time} \quad \textit{and} \quad O(\text{size}(d, k)) \textit{ space.}$$

Proof. According to Theorem 1, the maximal number of faces of low V_m equals size(d, m) and, in particular, $|V_m| = \text{reg}(d, m)$. Thus the individual steps of the algorithm have the following time complexity (d is considered as a constant):

Step 1.1. $O(\text{reg}(d, 1))$. Obvious.
Step 1.2. $O(T_{d+1}(\text{reg}(d, 1)))$. By definition.
Step 1.3. $O(\text{size}(d, 1))$. Due to the properties of an incidence lattice.
Step 2.1. $O(\text{size}(d, m-1))$. Each triangle t of low V_{m-1} is pushed onto the stack at most once, and constant time suffices for finding and processing t.
Step 2.2. $O(T_{d+1}(\text{reg}(d, m)))$. By definition.
Step 2.3. $O(\text{size}(d, m))$. By similar arguments as for Step 2.1.
Step 3.1. $O(\text{size}(d, k))$. Obvious.
Step 3.2. $O(\text{size}(d, k))$. Each vertex is calculated in $O(d)$ time.

Note that size(d, m) is a natural lower bound on $T_{d+1}(\text{reg}(d, m))$. Hence the time complexity of the algorithm is dominated by Steps 1.2 and 2.2 which yields the above formula.

The storage needed remains in $O(\text{size}(d, k))$, since the convex hull algorithms in [8]-[10] are space-optimal, the number of faces of conv V_k is in $O(\text{size}(d, k))$, and each face of conv V_k only is augmented with a constant amount of data. □

In the case $d = 2$, where $\text{reg}(2, k) < \text{size}(2, k) = O(k(|P| - k))$ and $T_3(r) = O(r \log r)$ hold (see [7] and [8], respectively), we obtain:

Corollary. *Algorithm CONSTRUCT DIAGRAM constructs the order-k Voronoi diagram $k - V(P)$ of a set P of n points in the plane in $O(k^2 n \log n)$ time and $O(k(n-k))$ space.*

In several applications of $k - V(P)$, the k-subset S of the k closest points in P for each cell cell(S) of $k - V(P)$ has to be available. In order to meet this requirement, the cell cell(S_0) which corresponds to vertex v_0 of low V_k is associated with S_0. For each facet f of $k - V(P)$, incident upon cell(S) and cell(S') say, the pointer in $I(k - V(P))$ between cell(S) and f (cell(S') and f) is associated with pair$(+e)$ (pair$(-e)$), if the edge $+e$ of low V_k is directed from $q(S)$ to $q(S')$. Since $S' = (S \cup \{p'\}) - \{p\}$, for pair$(+e) = (p', p)$, the desired k-subsets can be calculated and assigned to their cells in $O(k)$ time each, by scanning through the cells of $k - V(P)$ starting at cell(S_0).

4. Discussion and Extensions

The contributions of this paper fall into two parts: the geometric part establishes a duality between order-k Voronoi diagrams and convex hulls. This result, which is of independent interest, can be considered as a refinement of a result obtained by the author in [2] using quite different techniques, which concerns a more general class of Voronoi diagrams called power diagrams. Using this geometric background, a new algorithm for constructing $k-V(P)$ for a set P of n points in E^d is proposed and worked out in detail. The method is space-optimal for any d and (like Lee's [7] algorithm) provides the most time-efficient solution in E^2 for $k<\sqrt{n/\log n}$. (This seems to include the most interesting values of k since k often is considered as a constant in practice.) However, our method is somewhat simpler than the one in [7] since each of its iterative steps computes only one convex hull rather than computing various particular order-1 Voronoi diagrams (using the divide-and-conquer scheme of [11]). Our algorithm is a direct, but nevertheless nontrivial, generalization to order k of the algorithm in [3] which computes $1-V(P)$ in E^d via a convex hull in E^{d+1}. Note that if the points in P are given in integer coordinates then only convex hulls of points with integer coordinates have to be determined.

Our duality result applies particularly well to the construction of $k-V(P)$ if $k\le d$. In this case the set V_k of vertices of low Q_k can be derived directly from $1-V(P)$ using the lemma below rather than iteratively employing Lemma 2.

Lemma 3. *Let S be a k-subset of P for $1\le k\le d$. $q(S)$ is in V_k iff $f=\bigcap_{p\in S}\text{cell}(\{p\})$ is a $(d-k+1)$-face of $1-V(P)$.*

Proof. By Theorem 1, $q(S)\in V_k$ means that cell(S) in $k-V(P)$ has dimension d. That is, there exists a sphere σ in E^d with $S\subsetneq \text{int}\,\sigma$ and $P-S\subsetneq \text{ext}\,\sigma$ (by definition of $k-V(P)$). Since $|S|\le d$, the latter is equivalent to the existence of a sphere σ', centered at x say, with $S\subsetneq\sigma'$ and $P-S\subsetneq \text{ext}\,\sigma'$. But this is necessary and sufficient for x to lie in the relative interior of the $(d-k+1)$-face f of $1-V(P)$. □

Note that the k-subsets S and S' that correspond to two $(d-k+1)$-faces of $1-V(P)$ incident upon a common $(d-k)$-face are of the form $S'=(S\cup\{p\})-\{p'\}$. Thus V_k, for $1\le k\le d$, can be calculated in a straightforward manner by scanning through the $(d-k+1)$-faces of $1-V(P)$.

As an open problem we state the direct calculation of all k-subsets of P with a nondegenerate cell in $k-V(P)$, for general k. (The problem is clearly settled for $k\le d$ by Lemma 3. However, it is not likely that methods similar to the one used in its proof carry over to the case $k>d$.) By virtue of the results in this paper, a solution requiring $O(k(n-k)\log n)$ time for $P\subsetneq E^2$ would imply the first known optimal algorithm for constructing $k-V(P)$ in E^2.

It is worth mentioning that our construction strategies can be applied to the more general class of order-k power diagrams for n spheres in E^d (if each sphere has associated a nonempty cell in the order-1 power diagram; see [1] for properties of power diagrams). Moreover, each order-k power diagram is the order-$(n-k)$ power diagram for some set of spheres, so that we can efficiently construct the latter diagram for small k, starting with order $n-1$.

References

1. Aurenhammer, F. Power diagrams: properties, algorithms and applications. *SIAM J. Comput.* **16** (1987), 78–96.
2. Aurenhammer, F. A criterion for the affine equivalence of cell complexes in R^d and convex polyhedra in R^{d+1}. *Discrete Comput. Geom.* **2** (1987), 49–64.
3. Brown, K. Q. Voronoi diagrams from convex hulls. *Inform. Process. Lett.* **9** (1979), 223-228.
4. Chazelle, B., Edelsbrunner, H. An improved algorithm for constructing kth-order Voronoi diagrams. *IEEE Trans. Comput.* **36** (11) (1987), 1349-1354.
5. Edelsbrunner, H. Edge-skeletons in arrangements with applications. *Algorithmica* **1** (1986), 93-109.
6. Edelsbrunner, H., O'Rourke, J., Seidel, R. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.* **15** (1986), 341-363.
7. Lee, D. T. On k-nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Comput.* **31** (1982), 478-487.
8. Preparata, F. P., Hong, S. J. Convex hulls of finite sets of points in two and three dimensions. *Comm. ACM* (1977), 87-93.
9. Seidel, R. A convex hull algorithm optimal for point-sets in even dimensions. M.Sc. Thesis, Rep. 81-14, Dept. Comput. Sci., University of British Columbia, Vancouver (1981).
10. Seidel, R. Constructing higher-dimensional convex hulls at logarithmic cost per face. *Proc. 18th ACM Symp. STOC* (1986), 404-413.
11. Shamos, M. I., Hoey, D. Closest-point problems. *Proc. 16th. Ann. IEEE Symp. FOCS* (1975), 151-162.

Received November 24, 1986, and in revised form July 10, 1987.