A CONVEX HULL ALGORITHM

OPTIMAL

FOR POINT SETS IN EVEN DIMENSIONS

by

RAIMUND SEIDEL

A THESIS SUBMITTED IN PARTIAL FULLFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

We accept this thesis as conforming
to the required standard.

THE UNIVERSITY OF BRITISH COLUMBIA

September 1980

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date Sept 8th 1981

ABSTRACT

Finding the convex hull of a finite set of points is important not only for practical applications but also for theoretical reasons: a number of geometrical problems, such as constructing Voronoi diagrams or intersecting hyperspheres, can be reduced to the convex hull problem, and a fast convex hull algorithm yields fast algorithms for these other problems.

This thesis deals with the problem of constructing the convex hull of a finite point set in R^d. Mathematical properties of convex hulls are developed, in particular, their facial structure, their representation, bounds on the number of faces, and the concept of duality. The main result of this thesis is an $O(n\log n + n^{\lfloor (d+1)/2 \rfloor})$ algorithm for the construction of the convex hull of n points in R^d. It is shown that this algorithm is worst case optimal for even $d \geq 2$.

Table of Contents

Acknowledgement

"Ein herzliches Dankeschön" to David Kirkpatrick. I do not think I could find a better supervisor. I want thank Alan Mackworth for his constant encouragement and for being a very careful second reader. Thanks also to my fellow students, especially Jim Little, for their patience.

TO CHINU

I. Introduction

In the young field of computational geometry the convex hull problem has been one of the most studied problems. Simply stated, the issue is to find the smallest convex set containing a given point set. The importance of the convex hull problem is twofold. First, it has numerous applications in engineering, pattern recognition, and other fields. Second, it plays a central role in computational geometry, as a number of other geometric problems, such as the construction of Voronoi diagrams, the construction of the union or intersection of spheres, and other problems, can be transformed or reduced to the convex hull problem.

As early as 1972 Graham [5] presented an O(nlogn) convex hull algorithm for planar point sets. Subsequently several papers have been published addressing different variants of the problem. Jarvis [8] developed an algorithm whose running time depends linearly on the product of the number of input points and the number of points found on the convex hull, Preparata [12] designed an O(nlogn) real time algorithm, and Bentley and Shamos [1] presented a linear expected time algorithm for sets of n points drawn from a distribution for which the expected number of points on the hull is $O(n^p)$, $p<1$. Yao [15] tackled the complexity of the problem from the other side and proved that Ω(nlogn) time is necessary to identify the vertices of the convex hull even if tests involving quadratic functions of the

inputs are allowed. Thus the algorithms of Graham and Preparata are worst case optimal.

For point sets in 3-space Preparata and Hong [13] developed an O(nlogn) algorithm which is based on the divide and conquer paradigm. This algorithm is of course optimal by Yao's result.

Whereas the convex hull problem in 2 and 3-space seems to be fairly well understood, the situation is quite different for convex hulls of point sets in d-space, d>3. The only general algorithm published is the one by Chand and Kapur [4]. It relies on the so called gift wrapping principle and uses linear time to determine each facet of the convex hull. As the convex hull of n points in R^d can have $\Theta(n^{\lfloor d/2 \rfloor})$ facets [10], this implies that worst case time complexity of this algorithm is not better than $O(n^{\lfloor (d+2)/2 \rfloor})$.

In this thesis an incremental algorithm for the construction of the convex hull of n points in d-space is presented which improves on the algorithm of Chand and Kapur. It is shown that it has worst case time complexity $O(n\log n + n^{\lfloor (d+1)/2 \rfloor})$ and that this is optimal for even d.

Section 2 of this thesis deals with the basic mathematical properties of polytopes. There is a rich theory dealing with polytopes, but unfortunately it seems to have been ignored by computer scientists. In Section 3 a method for the intersection

of a polytope and a halfspace is developed on which the convex hull algorithm presented in Section 4 is based. In the last section the results are discussed and some open problems are presented. In particular, it is noted that the convex hull algorithm of this thesis yields an algorithm for the construction of Voronoi diagrams which is optimal for point sets in odd dimensions.

II. Convex Polytopes

This section covers the basic definitions and properties of multi-dimensional convex polytopes. There is an extensive theory and literature about these objects. The definitions and properties given in this section represent a bare minimum of facts pertinent to the content of the subsequent sections. For a more complete treatment of polytopes and related subjects the reader is referred to Gruenbaum's book [6].

In the following we will deal with the d-dimensional Euclidean space R^d, where d is an arbitrary positive integer. It is assumed that the reader is familiar with basic notions of linear algebra such as subspace, dimension of a subspace, scalar product, orthogonality, etc. Familiarity with basic topological notions such as open and closed sets is also assumed.

For $x,y \in R^d$, $\underline{\langle x,y \rangle}$ will denote the scalar product of x and y, and for a set $S \subset R^d$, $\underline{\text{int } S}$ will denote the interior of S under the topology induced by the Euclidean metric.

<u>Definition 2.1</u>:

Let $a \in R^d = (a \neq 0)$, and $c \in R$.

The set $\{x \in R^d \mid \langle x,a \rangle = c\}$ is called <u>a hyperplane</u>.

The set $\{x \in R^d \mid \langle x,a \rangle \leq c\}$ is called a <u>closed halfspace</u>.

It should be clear that closed halfspace could also be

defined using "≥" instead of "≤", and that it is a closed set in the topological sense.

<u>Definition 2.2</u>:

Let $a \in R^d$, let X be a linear subspace of R^d, and let k, $-1 \le k \le d$, be the dimension of X. (By convention, let the empty set be a subspace of dimension -1.)

$H = a+X = \{a+x \mid x \in X\} \subset R^d$ is called a <u>flat</u>. The <u>dimension of H</u>, dim(H), is k. A flat of dimension k is also called a <u>k-flat</u>.

Examples of flats are points (0-flat), lines (1-flat), and hyperplanes ((d-1)-flat).

<u>Definition 2.3</u>:

Let $S \subset R^d$. The <u>affine hull of S</u>, aff S, is the intersection of all flats containing S.

It easy to prove that the intersection of a family of flats is itself a flat. Thus the affine hull of any subset of R^d is a flat. For $S \subset R^d$ let dim S = dim(aff S).

<u>Definition 2.4</u>:

A set $S \subset R^d$ is called <u>convex</u> iff for any $x,y \in S$, $seg(x,y) = \{(1-r)x+ry \mid 0 \le r \le 1\} \subset S$.

Examples of convex sets are flats and halfspaces. Observe that

the intersection of a family of convex sets is a convex set. Furthermore the following holds:

Lemma 2.1:

A closed convex set $K \subset R^d$ is the intersection of all closed halfspaces containing K.

Proof: [6] 2.2.3. Q.E.D.

Definition 2.5:

Let $K \subset R^d$ be a convex set and $H = \{x \in R^d | \langle a,x\rangle = c\}$ a hyperplane.

H is called a supporting hyperplane of K if either $\inf\{\langle x,a\rangle | x \in K\} = c$ or $\sup\{\langle x,a\rangle | x \in K\} = c$.

We say, a supporting hyperplane H of K separates K and a point $p \in R^d - K$ if p is not contained in the same halfspace determined by H as K.

Definition 2.6:

Let K be a convex subset of R^d and k=dim K.

1) $F \subset K$ is called a face of K if either F=K or $F=\emptyset$ or there exists a hyperplane H supporting K, such that $K \cap H = F$.
2) A face F with dim F = m is call an m-face of K.
3) A (k-1)-face of K is called a facet of K, a 0-face is called a vertex, and a 1-face is called an edge of K.
4) If F is an m-face of K, G an (m+1)-face, and $F \subset G$, then we call F a subface of G and G a superface of F.
5) A face F of K is called proper, if $F \neq \emptyset$ and $F \neq K$.

Clearly, every face of a convex set is convex. Furthermore the following holds:

Lemma 2.2:

The intersection of a family of faces of a closed convex set K is a face of K.

Proof: [6] 2.4.10. Q.E.D.

Definition 2.7:

The convex hull of a set $A \subset R^d$, conv A, is the intersection of all closed convex sets containing A.

In view of Lemma 2.1 an alternative definition of conv A is the intersection of all closed halfspaces containing A. For every A conv A is closed and convex. The following lemma gives another equivalent definition for the convex hull.

Lemma 2.3:

Let P be the convex hull of a set A in R^d.

1) P comprises exactly all $x \in R^d$ which are expressible in the form $x = \sum_{i=0}^{d} c_i x_i$, where $x_i \in A$, $c_i \geq 0$, and $\sum_{i=0}^{d} c_i = 1$.

2) If A is finite, $A = \{x_1, \ldots, x_n\}$, then the relative interior of P comprises exactly all $x \in R^d$ which are expressible in the form $x = \sum_{i=1}^{n} c_i x_i$, where $c_i > 0$, and $\sum_{i=1}^{n} c_i = 1$.

Proof:

1) [6] 2.3.3, 2.3.5.

2) [11] page 11.

Q.E.D.

Definition 2.8:

A polytope is the convex hull of a finite set of points. A polytope P is called a d-polytope iff dim P = d.

Lemma 2.4:

Every face of a polytope is a polytope.

Proof: Consequence of [6] 3.1.4. Q.E.D.

Lemma 2.5:

Let P be a polytope in R^d and dim P = d.

1) The number of distinct faces of P is finite.

2) Every (d-2)-face of P is the intersection of exactly two facets.

3) Every k-face of P is the intersection of at least d-k facets of P.

Proof: [6] 2.6.3, 2.6.4, 3.1.6, 3.1.8. Q.E.D.

For a polytope P with dim P = d, we denote the number of i-faces (which is finite) by $f_i(P)$. By convention $f_{-1}(P)=f_d(P)=1$, and $f_i(P)=0$ for $i<-1$ and $i>d$.

Lemma 2.6:

If F is a face of a polytope P and G is a face of the polytope F, then G is also a face of P.

Proof: [6] 3.1.5. Q.E.D.

Definition 2.9:

Let P be a polytope. The k-skeleton of P, $skel_k$ P, is the set of all i-faces of P, $0 \leq i \leq k$.

Observe that the 1-skeleton of a polytope can be interpreted as a graph, where the 0-faces are the vertices and the 1-faces are the edges of the graph. Of some importance is the following:

Lemma 2.7:

The graph representing the 1-skeleton of a d-polytope is d-connected.

Proof: [6] 11.3.2. Q.E.D.

Definition 2.10:

The facial graph of a polytope P, FG(P), is a directed graph whose nodes correspond to the faces of P. FG(P) has an arc from the node corresponding to the face F_1 of P to the node corresponding to face F_2 iff F_1 is a subface of F_2.

Definition 2.11:

Two polytopes are said to be combinatorially equivalent iff their facial graphs are isomorphic.

Clearly the facial graph of a d-polytope P, FG(P), is an acyclic graph with one source and one sink which has d+2 levels, i.e. all directed paths from source to sink are of length d+1. Furthermore, as it is shown below, the converse of FG(P) (i.e. FG(P) with the direction of all arcs reversed) is realizable as the facial graph of some d-polytope P^*.

Definition 2.12:

The polar set A^* of $A \subset R^d$ is defined by

$A^* = \{y \in R^d | \langle x,y \rangle \leq 1 \text{ for all } x \in A\}$.

Lemma 2.8:

Let P be a d-polytope and $0 \in \text{int } P$.

1) P^* is a d-polytope with $0 \in \text{int } P^*$ and $(P^*)^* = P$.

2) If F is a k-face of P then

$$F' = \{y \in P^* | \langle x,y \rangle = 1 \text{ for all } x \in F\}$$

is a (d-k-1)-face of P^*.

3) The mapping ϕ defined by $\phi(F) = F'$ is a 1-1 inclusion reversing correspondence between the set of faces of P and the set of faces of P^*.

Proof: [6] 3.4. Q.E.D.

Definition 2.13:

Let P be a d-polytope and $0 \in \text{int } P$.

P^* is called the dual of P.

Lemma 2.8 implies that the facial graphs of a polytope and its dual are anti-isomorphic. Observe that if v is a vertex of P, with $0 \in \text{int } P$, then $\{x \in R^d | \langle x,v\rangle = 1\}$ is a supporting hyperplane of P^* and contains the facet v' dual to v. In other words, if $0 \in \text{int } P$, and $V = \{v_1,\ldots,v_n\}$ is the set of vertices of P, then $P^* = \bigcap \{H_i | 1 \le i \le n\}$, where $H_i = \{x \in R^d | \langle x,v_i\rangle \le 1\}$ and each of the n hyperplanes $\{x \in R^d | \langle x,v_i\rangle = 1\}$ contains a facet of P^*.

In the following we introduce some special polytopes and state some of their properties.

<u>Definition 2.14</u>:

1) A <u>k-simplex</u> is a k-polytope which is the convex hull of k+1 points in R^d, $d \ge k$.
2) P is called a <u>simplicial polytope</u>, iff all its proper faces are simplices.

It should be noted that a k-simplex has precisely $\binom{k+1}{i+1}$ i-faces, and that each i-face is an i-simplex itself. Futhermore, a simplex is combinatorially equivalent to its own dual.

<u>Definition 2.15</u>:

Let B be a (d-1)-polytope.

If c is a point in R^d-aff B, then $P = \text{conv}(B \cup \{c\})$ is called a <u>d-pyramid with basis B</u>.

The faces of P satisfy the following relationship with the faces of B.

<u>Lemma 2.9</u>:

Let P be a d-pyramid with basis B and apex c, and let F be an i-face of B.

1) F is also an i-face of P and $\mathrm{conv}(F \cup \{c\})$ is an (i+1)-face of P.
2) Every i-face of P is either also an i-face of B or it is of the form $\mathrm{conv}(G \cup \{c\})$, where G is an (i-1)-face of B.

<u>Proof</u>: [6] 4.2. Q.E.D.

Using Lemma 2.9, it easy to see how the facial graph of a pyramid P with basis B and apex c can be expressed in terms of the facial graph of B. FG(P) consists of two copies of FG(B). The nodes in one copy correspond to the faces of P which are also faces of B, the nodes in the other copy correspond to the faces of P of the form $\mathrm{conv}(G \cup \{c\})$, where G is a face of B. Each node in the first copy corresponding to a face F of B has an arc to the node in the second copy corresponding to the face $\mathrm{conv}(F \cup \{c\})$. Thus in graph theoretic terms, the underlying graph of FG(P) is the product graph $K_2 \times FG(B)$.

Observe, that a d-simplex is a d-pyramid whose basis is a (d-1)-simplex. Thus the underlying graph of the facial graph of a d-simplex is a (d+1)-cube. (For the graph theoretic terms see [7].)

Of considerable interest is the family of cyclic polytopes.

<u>Definition 1.16</u>:

Let M_d be the moment curve defined by $M_d(t) := (t, t^2, \ldots t^d)$, $t \in R$. The d-polytope formed by the convex hull of any $n>d$ distinct points on the moment curve M_d is called a <u>cyclic polytope</u> <u>$C(n,d)$</u>.

Cyclic polytopes are simplicial ([6] 4.7), and the combinatorial type of $C(n,d)$ is independent of the choice of the n points on M_d. That is, any two d-dimensional cyclic polytopes on n vertices are combinatorially equivalent. The importance of cyclic polytopes lies in their extremal nature.

<u>Lemma 2.10</u>:

1) If P is a d-polytope on n vertices, then $f_k(P) \le f_k(C(n,d))$ for all k.

2) The number of k-faces of a cyclic polytope $C(n,d)$ is $O(n^{\lfloor d/2 \rfloor})$. More specifically, for $0 \le k < d$

$$f_k(\ C(n,2s)\ ) = \sum_{i=1}^{s} \frac{n}{i}\binom{n-i-1}{i-1}\binom{i}{k-i+1} \qquad \text{for } d=2s$$

$$f_k(C(n,2s+1)) = \sum_{i=1}^{s} \frac{k+2}{i+1}\binom{n-i-1}{i}\binom{i+1}{k-i+1} \qquad \text{for } d=2s+1.$$

For $k=d-1$ these expressions evaluate to

$$f_{d-1}(C(n,d)) = \frac{n}{n-s}\binom{n-s}{s} \qquad \text{for } d=2s\ ,$$

$$f_{d-1}(C(n,d)) = 2\binom{n-s-1}{s} \qquad \text{for } d=2s+1.$$

<u>Proof</u>: 1) [10] , 2) [6] 9.6.1. Q.E.D.

The lemma above gives a bound on the number of faces of a d-polytope on n vertices. It should be noted that, by duality, the same bounds hold for the number of vertices of polytopes with n facets. However, for the sake of representation of polytopes, the set of faces is rather uninteresting because it does not explicitly express any of the geometric or combinatorial structure of a polytope. The facial graph of a polytope carries such information. Thus a bound on the size of a facial graph, i.e. the number of its nodes and arcs, is of interest.

Let N(P) denote the number of nodes in the facial graph of a polytope P (i.e. N(P) is the number of faces of P), let A(P) denote the number of arcs in FG(P) (i.e. the number of face - subface pairs in P), and let D(P) = A(P) + N(P).

We will give a bound on D(P) in terms of the number of vertices of P in two steps. First it will be argued that it suffices to consider only simplicial polytopes. Second, we will derive a bound for D(P), where P is a simplicial polytope.

<u>Lemma 2.11</u>:

Let P be a d-polytope, V the set of vertices of P, and $v \in V$. By a process called pulling, vertex v can be perturbed slightly to v', such that the d-polytope $P' = \mathrm{conv}(\ (V-\{v\}) \cup \{v'\}\ )$ has exactly the following k-faces for $0 \le k < d$:

(i) the k-faces of P which do not contain v;

(ii) the convex hulls of the type $\mathrm{conv}(\{v'\} \cup G^{k-1})$, where G^{k-1} is a (k-1)-face not containing v of a facet of P which does contain v.

Proof: [6] 5.2.2. Q.E.D.

Lemma 2.12:

If P^* is obtained from d-polytope P by successively pulling each of the vertices of P, then P^* is a simplicial d-polytope satisfying

1) $f_0(P^*) = f(P)$ and $N(P^*) \geq N(P)$,

2) $A(P^*) \geq A(P)$.

Proof:

1) [6] 5.2.4

2) We only need to show that pulling a vertex v of polytope P to v' to yield polytope P' does not decrease the number of face - subface pairs.

Let F be a k-face of P and G a subface of F for $1 \leq k < d$. We will show that there is a corresponding pair of faces F' and G' of P', such that G' is a subface of F'.

There are three cases to consider:

(i) F and G do not contain v:

then F and G are also faces of P' and G is a subface of F;

(ii) F contains v, G does not contain v:

F is not a face of P'; but G is a face of P' and by

Lemma 2.11 induces a face $F' = \mathrm{conv}(G \cup \{v'\})$, and G is a subface of F';

(iii) both F and G contain v:

F and G must have respective subfaces F^- and G^-, such that G^- is a subface of F^- and both F^- and G^- do not contain v. By Lemma 2.11 F^- and G^- induce faces $F' = \mathrm{conv}(F^- \cup \{v'\})$ and $G' = \mathrm{conv}(G^- \cup \{v'\})$ of P' such that G' is a subface of F'.

Q.E.D.

<u>Lemma 2.13</u>:

For any d-polytope P on n vertices $D(P) = O(n^{\lfloor d/2 \rfloor})$.

<u>Proof</u>:

Let P be a d-polytope on n vertices. By Lemma 2.12 $D(P) \le D(P^*)$ for some simplicial d-polytope P^* on n vertices. As every facet of P^* is a (d-1)-simplex, clearly $D(P^*) < f_{d-1}(P^*)*D(S_{d-1})$, where S_k denotes a k-simplex. As it was mentioned after Lemma 2.9, the underlying graph of $FG(S_{k-1})$ is a k-cube and therefore has 2^k vertices and $k*2^{k-1}$ edges. Thus $D(S_{d-1}) = (d+2)*2^{d-1}$.

Furthermore, by Lemma 2.10 $f_{d-1}(P^*) \le f_{d-1}(C(n,d))$.

For even d=2s, $f_{d-1}(C(n,d)) = \frac{n}{n-s}\binom{n-s}{s} \le \frac{n^s}{s!}$.

By a routine application of Stirling's approximation formula one can show that $D(S_{d-1}) = (s+1)*2^{2s} = O(s!)$. Hence $D(P) \le D(P^*) < f_{d-1}(C(n,d))*D(S_{d-1}) = O(n^s)$.

Using the same method, one can show that $D(P) = O(n^s)$ for odd $d=2s+1$.

Thus $D(P) = O(n^{\lfloor d/2 \rfloor})$ for any d-polytope on n vertices.

Q.E.D.

III. The Intersection of a Polytope and a Halfspace

In this section we consider in some detail the problem of intersecting a polytope and a halfspace. The efficient construction of such an intersection will be the main tool of the convex hull algorithm presented in the next section. We formally state the problem as follows:

> Given the facial graph of a polytope P, a halfspace H, and a vertex v of P not contained in H, construct the facial graph of the polytope $P' = P \cap H$.

The main result of this section asserts that this problem can be solved in time proportional to the amount of change from the facial graph of P to the facial graph of P'.

Throughout this section H denotes a closed halfspace in R^d defined by the hyperplane I, H^- denotes the other closed halfspace defined by I, i.e. $H^- = (R^d - H) \cup I$, and P is a d-polytope.

Definition 3.1:

A face F of P is called

a good face with respect to H, iff $F \subset H$ and $F \not\subset H^-$,

a cut face with respect to H, iff $F \subset H$ and $F \subset H^-$ (i.e. $F \subset I$),

a bad face with respect to H, iff $F \subset H^-$ and $F \not\subset H$, but no subface of F is contained in I,

a <u>touch face with respect to H</u>, iff $F \subset H^-$ and $F \not\subset H$, and one subface of F is contained in I, and

a <u>mixed face with respect to H</u>, iff $F \not\subset H^-$ and $F \not\subset H$.

In the following, when no confusion can arise, the phrase "with respect to H" will be omitted, and we will just talk about good faces, bad faces, etc. Observe that every nonempty face of P falls in exactly one of the five categories implied by Definition 3.1. Figure 3.1 illustrates these five different types of faces for the case of d=2.

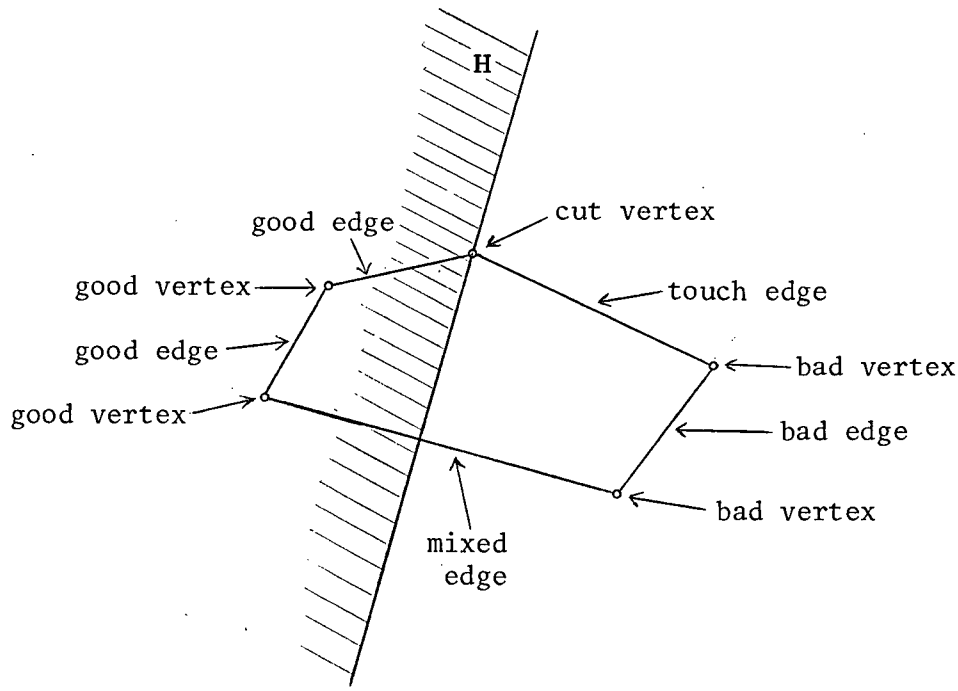

Figure 3.1: The five types of faces.

The following lemma gives a characterization of the type of a face of P in terms of the types of its subfaces or superfaces.

Lemma 3.1:

A face F of P is

(i) a good face, iff all its subfaces are good or cut faces,

(ii) a cut face, iff a superface of F is a touchface,

(iii) a bad face, iff all its subfaces are bad or touch faces,

(iv) a touch face, iff all but one subfaces are bad or touch faces,

(v) a mixed face, iff a subface of F is mixed or, with the exception of at least two, some subfaces of F are bad or touch faces.

Proof:

Follows straightforwardly from Definition 3.1. Q.E.D.

The next two lemmas give a characterization of the faces of $P' = H \cap P$ in terms of the faces of P, and a characterization of the face - subface pairs of P' in terms of face - subface pairs of P.

Lemma 3.2:

(i) If F is a good face or a cut face of P, then F is also a face of P'.

(ii) If F is a mixed k-face of P, then $F \cap H$ is a k-face of P' and $F \cap I$ is a (k-1)-face of P'.

(iii) If F is a bad face or a touch face of P, then F is <u>not</u> a face of P'.

(iv) Points (i) and (ii) yield all faces of P'.

<u>Proof</u>:

(i) and (iii) are trivial.

(ii) If F is a mixed k-face of P, then there is a supporting hyperplane X, such that $X \cap P = F$. Certainly, $X \cap P \cap H = F \cap H$, and X is a supporting hyperplane of $P' = P \cap H$.

Observe that $G = P \cap I$ is a facet of P'. $F \cap I$ is a subface of $P \cap H$ and $F \cap I = (F \cap H) \cap G$. Thus by Lemma 2.6 $F \cap I$ is a (k-1)-face of P'.

(iv) This follows as a consequence of Lemma 2.5 and the fact that $G = P \cap I$ is the only facet of P' which is not contained in a facet of P.

Q.E.D.

<u>Lemma 3.3</u>:

Let G be a face of P and F a subface of G.

(i) If F is a good face or a cut face of P and G is a good face or cut face of P, then F is a subface of G on the polytope P'.

(ii) If F is good and G is mixed, then F is a subface of $G \cap H$ on P'.

(iii) If G is mixed, then $G \cap I$ is a subface of $G \cap H$ on P'.

(iv) If F and G are mixed, then $F \cap H$ is a subface of $G \cap H$,

and $F \cap I$ is a subface of $G \cap I$ on P'.

(v) If G is mixed and F is a touch face, then $F \cap I$ is a subface of $G \cap I$ on P'.

(vi) Points (i) to (v) yield all face - subface pairs on P'.

Proof:

(i) to (v) follow straightforwardly from Definition 3.1 and Lemma 3.2.

(vi) follows from Lemma 3.2 and the fact that all possible types of face - subface pairs are considered.

Q.E.D.

The preceding lemmas justify the following algorithm, which constructs the facial graph of P' from the facial graph of P.

We assume that we are dealing with a version of the facial graph in which every node corresponding to a vertex has associated with it the coordinates of the vertex.

Algorithm 3.1:

Intersection of a d-polytope P and a halfspace H.

The algorithm takes as input the facial graph of P in which all nodes corresponding to vertices not in H (i.e. bad vertices) and their incident arcs have been removed. Furthermore, the set of bad eges of P, BAD_1, the set of touch edges of P, $TOUCH_1$, and the set of mixed edges of P, $MIXED_1$, are assumed to be known.

For $k=1,\ldots,d$ BAD_k, $TOUCH_k$, and $MIXED_k$ stand for the set of bad, touch, and mixed edges of P respectively. At the beginning of the algorithm $BAD_k = TOUCH_k = MIXED_k = \emptyset$ for all $k>1$.

For a face F of P, sub(F) denotes the set of subfaces of F (i.e. the arcs of the facial graph pointing to the node corresponding to F), super(F) denotes the set of superfaces of F (i.e. the arcs of FG(P) leaving the node corresponding to F), T-sub(F) stands for the set of subfaces of F which are touch faces, and M-sub(F) stands for the set of subfaces which are mixed faces. For a mixed face F of P, induced(F) denotes $F \cap I$, the "new" face of P' induced by F.

It is assumed that at the beginning of the algorithm T-sub(F) := M-sub(F) := induced(F) := $\emptyset$ for all k-faces with $k>1$.

For each edge $F \in MIXED_1$, it is assumed that a node corresponding to the vertex $v_F = F \cap I$ of P' has been created, that the coordinates of v_F have been computed, and that induced(F) = v_F, $v_F \in$ sub(F), super(v_F) = $\{F\}$, sub(v_F) = $\{E\}$, and $v_F \in$ super(E), where E denotes the node corresponding to the empty face.

```
begin
   L := Ø
   for k = 2 to d do
      begin
        1.  Determine all bad, touch, and mixed k-faces
        for all F in BAD_k-1 do
            for each G in super(F) do delete F from sub(G)
                                      insert G in L
                                      delete F from the
                                                     facial graph

        for all F in TOUCH_k-1 do
            for each G in super(F) do delete F from sub(G)
                                      insert F in T-sub(G)
                                      insert G in L

        for all F in MIXED_k-1 do
            for each G in super(F) do insert G in MIXED_k
                                      insert F in M-sub(G)

        for all F in L do
            If cardinality( sub(F) ) = 0 then insert F in BAD_k
            If cardinality( sub(F) ) = 1 then insert F in TOUCH_k
            If cardinality( sub(F) ) > 1 then insert F in MIXED_k
            delete F from L

        2.  Establish the new face - subface relationships and
        create the new faces formed by the intersection of a
        mixed face and I.
        for each F in MIXED_k do
            create a (k-1)-face G
            induced(F) := G
            insert G in sub(F)
            super(G) := {F}
            for all X in M-sub(F) do
                 insert induced(X) in sub(G)
                 insert G in super(induced(X))
            for all X in T-sub(F) do
                 Let Y be the only element in sub(X)
                 (Y is a cut face)
                 insert Y in sub(G)
                 insert G in super(Y)

        3.  Cleanup
        for each F in MIXED_k-1 do
            induced(F) := M-sub(F) := T-sub(F) := Ø
        for each F in TOUCH_k-1 do
            Let Y be the only element in sub(F)
            delete F from super(Y)
            delete F from the facial graph
   end (of the k-loop)
```

4. Final Cleanup
if $MIXED_d \neq \emptyset$ then
    ($P \cap H$ is a d-polytope)
    $MIXED_d$ contains exactly one element, the node corresponding to P.
    induced(P) := M-sub(P) := T-sub(P) := $\emptyset$
if $TOUCH_d \neq \emptyset$ then
    ($P \cap H$ is a (d-1)-polytope)
    $TOUCH_d$ contains exactly one element, the node corresponding to P. P has exactly one facet left: the face $F = P \cap I$.
    delete P from super(F)
    delete P from the facial graph
end (of Algorithm 3.1).

For a polytope P and a halfspace H, let D(P,H) denote the number of bad faces and touch faces of P with respect to H plus the number of arcs of the facial graph incident to nodes corresponding to such faces. Furthermore, let M(P,H) denote the number of mixed faces of P plus the number of arcs between nodes corresponding to such faces. Observe, that D(P,H) is the number of nodes and arcs deleted from the facial graph of P. Also note, the M(P,H) = O(N), where N is the size of the facial graph of the facet $P \cap I$ of P'. The following holds:

Lemma 3.4:

Algorithm 3.1 correctly determines the facial graph of $P' = P \cap H$ in time O(D(P,H) + M(P,H)).

Proof:

Correctness follows from Lemmas 3.1 to 3.3. With respect to the time bounds observe the following. The algorithm considers only nodes and arcs which are either deleted from the facial graph or

are related to the insertion of a new node or arc. Only a constant amount of time needs to be spent on each of these nodes and arcs if appropriate pointer structures to manipulate the various sets are used.

Q.E.D.

Algorithm 3.1 assumes that the nodes corresponding to bad vertices of P and their incident arcs have been removed from the facial graph. It also assumes that the sets of bad edges, touch edges, and mixed edges are known. The next lemma shows that the subgraph of $skel_1$ P induced by these vertices and edges is connected. This implies that if only one bad vertex of P is known, all the vertices and edges mentioned above can be found by a depth first search, which traverses only bad, touch, and mixed edges of P. Thus this search can be performed in time proportional to the number of these edges plus the number of bad vertices.

<u>Lemma 3.5</u>:

Let P be a d-polytope and H a halfspace as specified at the beginning of this section.

The graph formed by the edges and vertices of P which are not contained in H is connected.

<u>Proof</u>:

Let H^- be $(R^d - H) \cup I$.

Clearly $P^- = P \cap H^-$ is a d-polytope and $F = P \cap I$ is a facet of P^-. The bad, touch, and mixed edges of P correspond to the

edges of P^- which are not contained in F. Similarly, the bad vertices of P are the vertices of P^- not contained in F. Thus we only need to show that the graph formed by the vertices and edges of the polytope P^- which are not contained in the facet F is connected. But this is implied by the following Lemma 3.6.

Q.E.D.

<u>Lemma 3.6</u>:

Let P be a d-polytope, d>1, and let F be a facet of P.
The graph formed by the vertices and edges of P which are not contained in F is connected.

<u>Proof</u>:

We have to show that for each pair v,w of vertices of P not contained in F there is a path from v to w containing no vertices or edges of F.

Induction on the dimension d:

(i) The statement is obviously true for d=2. The removal of an edge from a polygon leaves a chain of edges.

(i) Assume the statement is true for every (d-1)-polytope.

We consider three cases:

a) v and w lie both on a facet G does not intersect F:
There must be a path from v to w on G because by Lemma 2.7 the graph formed by $skel_1$ G is (d-1)-connected.

b) v and w lie both on a facet G which intersects F:
The intersection of G an F must be contained in a facet of G. Hence there is a path from v to w containing no edges

or vertices of F by the inductive assumption.

c) v and w lie on different facets G and G' of P:

Because the dual of P is a d-polytope the graph formed by its i-skeleton is d-connected. Therefore there is a sequence of facets of P, $G_o,\ldots,G_k$, such that $G=G_0$, $G'=G_k$, F is not contained in the sequence, and for $0<i\leq k$, G_i and G_{i-1} share vertices.

Thus, by virtue of a) and b) one can compose a path from v to w containing only vertices and edges of the facets G_i and not containing any vertex or edge of F.

Q.E.D.

Theorem 3.1:

Given the facial graph of a d-polytope P, a closed halfspace H defined by the hyperplane I, and a vertex v of P not contained in H, the facial graph of the polytope $P' = P \cap H$ can be constructed in time $O(D(P,H) + M(P,H))$, where D(P,H) denotes the number of nodes and edges deleted from FG(P), and M(P,H) denotes the size of the facial graph of the facet of P' which is contained in I.

Proof:

As a consequence of Lemma 3.5, a depth first search starting at vertex v through the graph formed by $skel_1$ P can be used to condition the facial graph of P to serve as an appropriate input for Algorithm 3.1. Furthermore, the depth first search also yields the sets of bad, touch, and mixed edges of P, so that Algorithm 3.1 can be applied to render the facial graph of P'.

The time necessary to perform the depth first search is proportional to the number of bad, touch, and mixed edges of P. This number is certainly less than D(P,H) + M(P,H). Algorithm 3.1 takes time O(D(P,H) + M(P,H)) by Lemma 3.4.

Q.E.D.

IV. The Convex Hull Algorithm

This section contains the main result of this thesis: a general convex hull algorithm which is worst case optimal for point sets in even dimensions. The main idea of the algorithm is to construct the convex hull of a point set incrementally after the set has been presorted.

Before we can describe any details of the algorithm, we have to settle the issue of representation. By definition the convex hull of a finite point set is a polytope. We will represent a polytope P by an augmented version of its facial graph FG(P). The modifications of FG(P) are as follows:

1) Each node of FG(P) which corresponds to a vertex of P has associated with it the coordinates of the vertex.
2) Each node of FG(P) which corresponds to a facet F of P has associated with it a vector u_F which is orthogonal to F.

As shown in Lemma 2.13, the size of the augmented facial graph of a d-polytope P = conv S, where S contains n points, is $O(n^{\lfloor d/2 \rfloor})$.

Let us now turn our attention to the concept of duality. Recall, that if the direction of all arcs of a facial graph FG(P) are reversed, the resulting graph is the facial graph of some polytope P^*, the dual of P. We argue that under certain conditions this is even true for an augmented facial graph.

Assume that the interior of a polytope P contains the origin, and let F be a facet of P. By Lemma 2.8, the vertex F' of P^* dual to F is the set $\{y \in P^* | \langle x,y\rangle = 1 \text{ for all } x \in F\}$ which contains just one element, the vector u_F normal to F which is scaled and oriented in such a way that $\langle x,u_F\rangle = 1$ for all x in F, and $\langle x,u_F\rangle \leq 1$ for all x in P. We call such a u_F the <u>normalized complement</u> of F. If in the augmented facial graph of a polytope P with $0 \in \text{int } P$ the vector associated with each node corresponding to a facet F is the normalized complement of F, then FG(P) with its arcs reversed is the augmented facial graph of P^*. In other words, such a graph can be interpreted in two ways: as the augmented facial graph of P, or as the augmented facial graph of P^*.

How can we make use of this duality in a convex hull algorithm? We exploit duality by reducing the convex hull problem to an intersection problem. Recall, that by the definition of polarity and Lemma 2.8, if P = conv V and $0 \in \text{int } P$, then $P^* = \cap\{H_v | v \text{ in } V\}$, where $H_v = \{x \in R^d | \langle x,v\rangle \leq 1\}$. It follows, that for any $q \in R^d$, $(\text{conv}(P \cup \{q\}))^* = (\text{conv}(V \cup \{q\}))^* = (P^* \cap H_q)$. As our representation of polytopes does not distinguish between duals, this implies that the problem of finding the convex hull of $P \cup \{q\}$ can be reduced to the problem of intersecting P^* with the halfspace H_q.

In the previous section we presented an algorithm which constructs such an intersection. However, this algorithm makes

the assumptions, that $P^* \cap H_q \neq P^*$, and that a bad vertex of P^*, i.e. a vertex of P^* which is not a vertex of $P^* \cap H_q$, is known. In the following we show how these assumptions can be met in an incremental convex hull algorithm.

<u>Definition 4.1</u>:

Let $p,q \in R^d$, $p=(p_1,\ldots p_d)$, $q=(q_1,\ldots q_d)$.

p is said to be <u>lexicographically smaller</u> than q, $p <_L q$, if $p_1 < q_1$, or if $p_1 = q_1$ and $(p_2,\ldots p_d) <_L (q_2,\ldots q_d)$.

Observe, that a set of n points in R^d can be sorted into lexicographical order in time $\Theta(n\log n)$. The next lemma draws a fundamental connection between lexicographical order and convex hulls.

<u>Lemma 4.1</u>:

Let $S \subset R^d$ be a finite set of distinct points, let $P = \text{conv } S$, and let p be the maximum element of S under the lexicographical ordering. Furthermore, let q be a point in R^d with $p <_L q$, and let $Q = \text{conv}(S \cup \{q\})$.

1) p is a vertex of P and $q \notin P$.

2) There is a facet of P which contains p and which is not a facet of Q.

<u>Proof</u>:

First note that it is always possible to rotate the coordinate system such that the lexicographical ordering of $S \cup \{q\}$ is

preserved but all the points in $S \cup \{q\}$ differ in their first coordinate.

We assume that such a rotation has been applied. We want to make it clear however, that this assumption is made only for the sake of simplicity of the proof. In the algorithm to be presented such a rotation never need be performed.

1) Let $u = (1,0,\ldots,0)$. By our assumption all elements of S different from p have a strictly smaller first coordinate. Hence $\langle x-p,u\rangle < 0$ for all $x \in S$, $x \neq p$, and as a consequence of Lemma 2.3, $\langle x-p,u\rangle < 0$ for all $x \in P$, $x \neq p$. As $\langle p-p,u\rangle = 0$, $H = \{x \in R^d \mid \langle x-p,u\rangle = 0\}$ is a supporting hyperplane of P, $P \cap H = \{p\}$, and p is a vertex of P.

   By assumption q has a strictly greater first coordinate than p. Thus $\langle q-p,u\rangle > 0$ and therefore H separates q from P, and $q \notin P$.

2) Let $A = \{F_1,\ldots,F_k\}$ be the set of facets of P which contain p. By Lemma 2.5 $k \geq d$. For $i=1,\ldots,k$ let a_i be a vector orthogonal to F_i, oriented such that $\langle x-p,a_i\rangle \leq 0$ for all x in P. Note, that by duality and Lemma 2.3 $u = \sum_{i=1}^{k} c_i a_i$, where $c_i \geq 0$ for all i.

   As $p <_L q$ and as their first coordinate are assumed to be different, $\langle q-p,u\rangle > 0$. But $\langle q-p,u\rangle = \langle q-p,\sum_{i=1}^{k} c_i a_i\rangle = \sum_{i=1}^{k} c_i \langle q-p,a_i\rangle$. As all c_i are non negative, $\langle q-p,a_j\rangle > 0$ for some j, $1 \leq j \leq k$. But then clearly F_j cannot be a facet of $Q = \mathrm{conv}(P \cup \{q\})$.

Q.E.D.

The following corollary is the dual formulation of Lemma 4.1.

Corollary 4.1:

Let $S \subset R^d$ be a finite set of distinct points, let $P = \text{conv } S$, and let $0 \in \text{int } P$. Let p be the maximal element of S under the lexicographical ordering and let q be a point in R^d with $p <_L q$.

1) The hyperplane $\{x \in R^d | \langle x,p\rangle = 1\}$ contains a facet p' of P^*, and $P^* \cap H_q \neq P^*$, where $H_q = \{x \in R^d | \langle x,q\rangle \leq 1\}$.
2) There is a vertex of P^* contained in the facet p' which is not a vertex of $P^* \cap H_q$.

Proof:

Follows immediately from Lemma 4.1 and duality. Q.E.D.

We now have all the tools needed to specify the algorithm for the construction of the convex hull of a finite point set.

Algorithm 4.1: Construction of a d-dimensional convex hull.

The algorithm takes as input a set $S \subset R^d$ of n distinct points. It outputs the augmented facial graph of conv S as it is specified at the beginning of this section.

1. Sort S into lexicographical order.

   Having sorted S, we can write $S = \{s_1,\ldots,s_n\}$, where for $1 \leq i < n$, $s_i <_L s_{i+1}$.

   For $1 \leq j \leq n$ let $S_j = \{s \in S | s <_L s_j\}$, let $P_j = \text{conv } S_j$, and let $P_{n+1} = \text{conv } S = P$.
2. Construction of an initial (d-1)-polytope.

   Let k, $d \leq k < n$, be such that $\dim S_k = d-1$, but dim

$S_{k+1} = d$.

Embed aff S_k in R^{d-1} and inductively construct the facial graph of the (d-1)-polytope $P_k = \text{conv } S_k$.

(By convention let the facial graph of the (-1)-polytope (the empty set) be a graph consisting of one node.)

By Definition 2.15 $P_{k+1} = \text{conv}(P_k \cup \{s_k\})$ is a d-pyramid with basis P_k and apex s_k. Construct the facial graph of P_{k+1} from $FG(P_k)$ as specified after Lemma 2.9.

Translate all points of S, such that the origin is contained in the interior of P_{k+1}. (Lemma 2.3 characterizes the interior points of a polytope.)

Associate with each node of $FG(P_{k+1})$ corresponding to a vertex of P_{k+1} the coordinates of this vertex, and associate with each node of $FG(P_{k+1})$ corresponding to a facet P_{k+1} the the normalized complement of this facet.

3. Insert the remaining points.

For i=k+1 to n+1 do

Construct $P_{i+1} = \text{conv}(P_i \cup \{s_i\})$.

As the origin is contained in the interior of P_i, this can be done by intersecting P_i^* and $H_i = \{x \in R^d | \langle x, s_i \rangle \leq 1\}$.

By Corollary 4.1 $P_i^* \cap H_i \neq P_i^*$, and one of the vertices of P_i^* contained in the facet s'_{i-1} of P_i^* dual to the vertex s_{i-1} of P_i is a bad vertex of P_i^* with respect to H_i.

Interpret $FG(P_i)$ as the facial graph of P_i^* and find a vertex contained in the facet s'_{i-1} of P_i^* which is not contained in H_i.

Apply Theorem 3.1 to construct the augmented facial graph of $P^*_{i+1} = P^*_i \cap H_i$. By duality this graph can also be interpreted as the facial graph of P_{i+1}.

4. Cleanup

Undo the translation of step 2 applied to the points of S.

<u>Theorem 4.1</u>:

Let $S \subset R^d$ be a set of n distinct points with dim S = d > 1. Algorithm 4.1 correctly determines the augmented facial graph of conv S in time $O(n^{\lfloor (d+1)/2 \rfloor})$. This is worst case optimal for even d.

<u>Proof</u>:

The correctness of the algorithm follows from Theorem 3.1, duality, and Corollary 4.1.

For the time bound consider the following:

The sort in step 1 clearly requires time O(nlogn).

Step 2:

The k can be found in O(k) time.

By induction, the facial graph of P_k can be constructed in time $O(k^{\lfloor d/2 \rfloor})$. The augmented facial graph of the pyramid P_{k+1} can then clearly also be constructed in time $O(k^{\lfloor d/2 \rfloor})$. Using Lemma 2.3, the translation required can be determined in O(k) time, and it can be applied to all points of S in linear time. Furthermore, the appropriate new values can be associated with the nodes of $FG(P_{k+1})$ corresponding to vertices and facets of P_{k+1} in time $O(k^{\lfloor d/2 \rfloor})$.

Step 3:

By Lemma 2.10 and duality, the facet s'_{i-1} of P_i^* can contain at most $O(i^{\lfloor (d-1)/2 \rfloor})$ vertices. Using the facial graph of P_i^*, all these vertices, and in particular a bad vertex with respect to H_i, can be found in time $O(i^{\lfloor (d-1)/2 \rfloor})$.

Applying Theorem 3.1, it takes time $O(M(P_i^*,H_i) + D(P_i^*,H_i))$ to intersect P_i^* and H_i. As remarked before Lemma 3.4, $M(P_i^*,H_i)$ is proportional to the size of the facial graph of the newly created facet s'_i of P_{i+1}^*. Thus by Lemma 2.13 $M(P_i^*,H_i)$ is $O(i^{\lfloor (d-1)/2 \rfloor})$.

Observe, that $D(P_i^*,H_i)$ is proportional to the number of nodes and arcs deleted from $FG(P_i^*)$ when P_i^* is intersected with H_i. As only $O(M(P_i^*,H_i)) = O(i^{\lfloor (d-1)/2 \rfloor})$ faces are created for every $i>k$,

$$\sum_{i=k+1}^{n+1} D(P_i^*,H_i) < O(k^{\lfloor d/2 \rfloor}) + \sum_{i=k+1}^{n+1} O(i^{\lfloor (d-1)/2 \rfloor}) = O(n^{\lfloor (d+1)/2 \rfloor}).$$

Thus $\sum_{i=k+1}^{n+1} O(M(P_i^*,H) + D(P_i^*,H_i)) = O(n^{\lfloor (d+1)/2 \rfloor})$ is the total time needed for step 3.

As step 4 can clearly be performed in linear time, the total worst case time complexity of the entire algorithm is $O(n\log n + n^{\lfloor (d+1)/2 \rfloor})$.

For even $d > 2$ this is optimal because by Lemma 2.13 the size of the description of the facial graph of conv S can be $O(n^{\lfloor d/2 \rfloor})$.

For $d = 2$ this is optimal because there is an $\Omega(n\log n)$ lower bound for the construction of the convex hull of a planar point set [15]. Q.E.D.

V. Conclusion

The main result of this thesis is a convex hull algorithm with $O(n\log n + n^{\lfloor (d+1)/2 \rfloor})$ worst case time complexity. In the formulation of the algorithm the concept of duality is used extensively. It should be noted however, that it seems possible to reformulate this convex hull algorithm without the use of dualization. Gruenbaum's ([6], p.78) characterization of the faces of the polytope $\mathrm{conv}(P \cup \{v\})$ in terms of the faces of the polytope P could be used for this purpose.

In Theorem 4.1 it is claimed that the convex hull algorithm presented in this thesis is worst case optimal for point sets in even dimensions. But the convex hull problem could also be formulated in a totally different way: given a set $S \subset R^d$ of n points, identify the points of S which are vertices of conv S. In this case the size of the facial graph of conv S is not a lower bound for this problem any more. Therefore it is possible that there is a solution for this problem whose worst case time complexity is better than $O(n\log n + n^{\lfloor (d+1)/2 \rfloor})$. For the planar case however, Yao [15] proved that this variant of the problem has still a lower bound of $\Omega(n\log n)$.

Finding an optimal convex hull algorithm for point sets was one of the major open problems stated in the work of Brown [3]. He showed that a number of geometrical problems could be reduced to the convex hull problem by the use of geometric transformation. In particular, he showed that constructing the

Voronoi diagram of a set $S \subset R^d$ is equivalent to contructing the convex hull of a set $S' \subset R^{d+1}$, where all points of S' lie on a hypersphere. As Klee [9] showed, that the description of a Voronoi diagram in R^d can be of size $O(n^{\lfloor (d+1)/2 \rfloor})$, the convex hull algorithm presented in this thesis also yields a Voronoi diagram algorithm which is optimal for point sets in odd dimension. Rather recently, two algorithms for Voronoi diagrams in arbitrary dimensions ([2],[14]), were published, which are based on a similar incremental approach as the convex hull algorithm of this thesis. However, the subquadratic time bounds claimed in these papers seem to be based on assumptions about the distribution of the input points and clearly cannot be worst case bounds by Klee's lower bound result.

A very striking point of the main result in this thesis is the fact that the convex hull algorithm is optimal for even dimensions, whereas this cannot be shown for odd dimensions. Naturally the question arises whether there is a convex hull algorithm for point sets in odd dimensions greater than three with worst case time complexity $O(n^{\lfloor d/2 \rfloor})$, which is a trivial lower bound of the problem, as the facial graph of the convex hull of n points can have this size. It is interesting to note that even for the three dimensional case no incremental algorithm with optimal, i.e. O(nlogn), worst case time complexity is known even if a presort is allowed. This seems to suggest that if there is an $O(n^{\lfloor d/2 \rfloor})$ algorithm for odd dimensions, it will have to use an approach entirely different from the one taken by the algorithm presented in this thesis.

REFERENCES

[1] Bentley, J.L. and Shamos, M.I.,
Divide and Conquer for Linear Expected Time,
Info. Proc. Letters 7, 2, Feb. 1978, pp.87-91.

[2] Bowyer, A.,
Computing Dirichlet Tessellations,
The Computer Journal, vol. 24, no.2, 1981, pp.162-166.

[3] Brown, K.Q.,
Geometric Transforms for Fast Geometric Algorithms,
Ph.D. thesis, Dept. of Comp. Sci., Carnegie Mellon Univ., Dec. 1979.

[4] Chand, D.R. and Kapur, S.S.
An Alorithm for Comvex Polytopes,
J.ACM 17, 1, Jan. 1970, pp.78-86.

[5] Graham, R.,
An Efficient Algorithm for Determining the Convex Hull of a Planar Set,
Info. Proc. Letters 1, 4, June 1972, pp.132-133.

[6] Gruenbaum, B.,
Convex Polytopes,
Wiley Interscience, 1967.

[7] Harary, F.,
Graph Theory,
Addison - Wesley, 1969.

[8] Jarvis, R.A.,
On the Identification of the Convex Hull of a Finite Set of Points in the Plane,
Info. Proc. Letters 2, 1973, pp.18-21.

[9] Klee, V.,
On the Complexity of d-dimensional Voronoi Diagrams,
Arch. Math., vol. 34, 1980, pp.75-80.

[10] McMullen, P.,
The Maximum Number of Faces of a Convex Polytope,
Mathematika 17, 1970, pp.179-184.

[11] McMullen, P. and Shephard, G.C.,
Convex Polytopes and the Upper Bound Conjecture,
London Mathl Society Lecture Note Series 3, Cambridge Univ. Press, 1971.

[12] Preparata, F.,
An Optimal Real-Time Algorithm for Planar Convex Hulls,
C.ACM 22, 7, July 1979, pp.402-405.

[13] Preparata, F. and Hong, S.J.,
Convex Hulls of Finite Sets of Points in Two and Three Dimensions,
C.ACM 20, 7, 1977, pp.87-93.

[14] Watson, D.F.,
Computing the n-dimensional Delaunay Tessellation with Application to Voronoi Polytopes,
The Computer Journal, vol. 24, no.2, 1981, pp.167-172.

[15] Yao, A.,
A Lower Bound to Finding Convex Hulls,
Rep. STAN-CS-79-733, Dept. of Comp. Sci., Stanford Univ., April 1979.