ANALYSIS OF A SIMPLE YET EFFICIENT CONVEX HULL ALGORITHM

Mordecai Golin
Robert Sedgewick

CS-TR-130-87

November 1987

Analysis of a Simple Yet Efficient Convex Hull Algorithm

Mordecai Golin[1]
Department of Computer Science
Princeton University

Robert Sedgewick[1]
Department of Computer Science
Princeton University

November 1987

Abstract.

This paper is concerned with a simple, rather intuitive preprocessing step that is likely to improve the average-case performance of any convex hull algorithm. For n points randomly distributed in the unit square, we show that a simple linear pass through the points can eliminate all but $O(\sqrt{n})$ of the points by showing that a simple superset of the remaining points has size $c\sqrt{n} + o(\sqrt{n})$. We give a full implementation of the method, which should be useful in any practical application for finding convex hulls. Most of the paper is concerned with an analysis of the number of points eliminated by the procedure, including derivation of an exact expression for c. Extensions to higher dimensions are also considered.

0. Introduction.

Convex hull algorithms can be seen as performing two separate tasks: The first is identifying those points actually on the hull, the second is eliminating those inside of it. Alternatively these two tasks can be seen as representative of two different approaches towards finding convex hulls. As described in [PS], "Giftwrapping" employs the first approach — it finds just those points that are on the hull and never identifies the rest; "Quickhull" concentrates on point elimination — finding the hull points is almost peripheral; "Graham's Scan" can be seen as balancing the two approaches, now finding a hull point, now eliminating an internal one.

In this paper we concentrate on point elimination: we present a linear-time preprocessing algorithm (similar to one step of Quickhull) that identifies and eliminates nearly all the points internal to the hull, under suitable assumptions. This algorithm is easily programmed and simple to modify for use in space of any dimension. Its output can then be fed into any convex hull routine. Most convex hull routines will run much more efficiently on few points than on many, so total running time is reduced. We are able to derive precise analytic results to substantiate these claims.

Section 1 presents the preprocessing algorithm. A Pascal implementation is given along with a worked example. In section 2 we analyze the expected number of points remaining after running the algorithm. Our probabilistic assumption is that the points are uniformly distributed in the unit square. Our main result is that $E(S) \leq c\sqrt{n}$ where S is the number of points remaining after running the algorithm on n points and c is a small constant less than 8. Section 3 discusses the combined behavior of the preprocessing routine followed by some standard convex hull algorithms. It also compares actual results to those predicted by the

[1] This work supported in part by NSF Grant DCR-8605962 and Office of Naval Research Grant N00014-87-K-0460.

mathematical analysis. In section 4 we discuss how to extend the algorithm and its analysis to space of any dimension. The result we state is that $E(S) = O(n^{(d-1)/d})$ where d is the index of the dimension. We finish by sketching an intuitive approach to understanding the algorithm's dynamics.

1. The Algorithm.

The problem of finding the convex hull of a set of n points $p_i = (x_i, y_i) \quad i = 1, \ldots, n,$ in $\Re^2$ has been extensively studied and many elegant algorithms exist for its solution. In addition to these there is also a very simple heuristic attributed to to W.F. Eddy and R.W. Floyd [Ed] [Se] for improving the performance of any such algorithm. Pick any four points from the p_i. These points are the vertices of a quadrilateral – in the degenerate case a triangle or line segment – Q. By its definition Q is contained within the convex hull of $p_1, \ldots, p_n$. We can therefore eliminate all points that are inside Q without discarding any information essential to the construction of the convex hull (figure 1). Our goal in this section is to identify a computationally simple method of choosing the four points such that after the elimination step only a few of the p_i will, on average, remain.

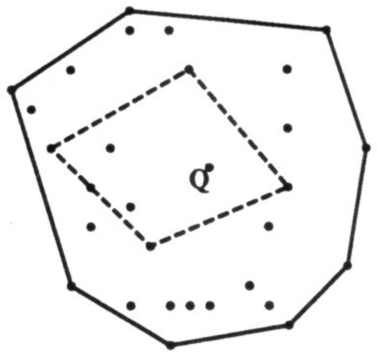

Figure 1.

Specifically, suppose that $p_1, \ldots, p_n$ are independent identically distributed random variables each of which is uniformly distributed in the unit square. In section 2 we will prove that the following choice of points gives us good average case behavior: Pick the four points that minimize the four functions $\pm x \pm y$. Equivalently, these are the two points that minimize the functions $x \pm y$ along with the two that maximize the same functions. Note that this definition is not well defined since there might be two points that minimize any of the functions. To correct this we will specify the points with the lowest indices that minimize the functions. Thus the four points (figure 2a) are

$$\begin{aligned}
q_1 = (x_1^q, y_1^q) = p_k \qquad & k = \min_j\{\forall i : +x_j + y_j \leq +x_i + y_i\} \\
q_2 = (x_2^q, y_2^q) = p_k \qquad & k = \min_j\{\forall i : +x_j - y_j \leq +x_i - y_i\} \\
q_3 = (x_3^q, y_3^q) = p_k \qquad & k = \min_j\{\forall i : -x_j + y_j \leq -x_i + y_i\} \\
q_4 = (x_4^q, y_4^q) = p_k \qquad & k = \min_j\{\forall i : -x_j - y_j \leq -x_i - y_i\}.
\end{aligned}$$

The next step is to eliminate those p_i inside Q. In reality the problem of deciding whether a point is inside an arbitrary quadralateral is nontrivial because of round-off error. For this reason we modify our elimination criteria somewhat. Instead of disposing of those points inside Q we find a large rectangle $R \subseteq Q$ and only dispose of those inside R. Since these points are also inside Q we lose no essential information. If Q is not degenerate there is a nice geometric method for finding a good R. Minimizing the function $x + y$ can be thought of as sweeping in a 45° line from the lower left corner of infinity until it hits some p_i which will then be q_1. Similarly minimizing $x - y$ sweeps in a line from the upper left hitting q_2, $-x + y$ sweeps in from the upper right hitting q_3, and $-x - y$ sweeps in from the lower right hitting q_4. By this reasoning the leftmost x-coordinate of R should be $\max(x_1^q, x_2^q)$ since these are the points encountered by sweeping in from the left. The rightmost x-coordinate will be the minimum of the two x-coordinates on the right, $\min(x_3^q, x_4^q)$ (figure 2b).The y boundaries of R are found in the same way and thus

$$R \;= \{(x, y) : \quad \max(x_1^q, x_2^q) < x < \min(x_3^q, x_4^q) \quad \max(y_1^q, y_3^q) < y < \min(y_2^q, y_4^q)\}.$$

(If Q is degenerate then R is an empty set (either its x or y range has nonpositive length) and thus none of the p_i are inside of it and the algorithm is still well defined.)

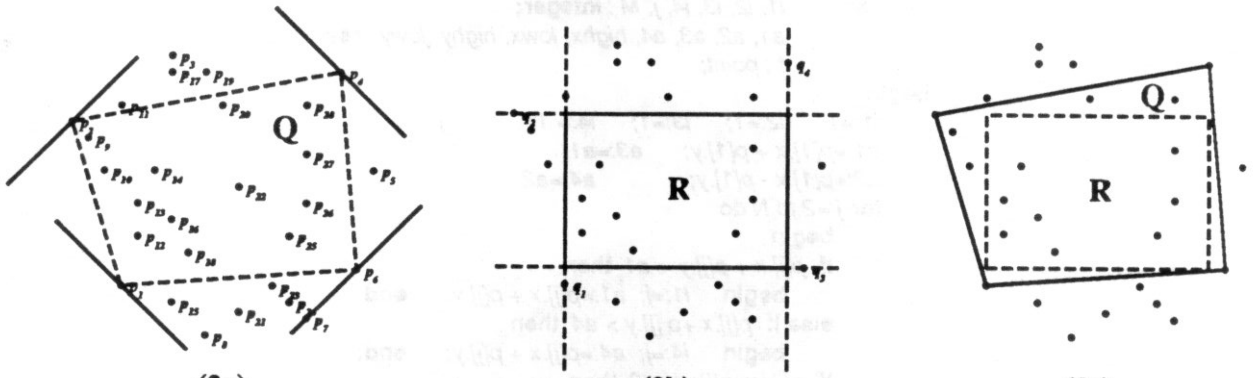
(2a) (2b) (2c)

Figure 2. (2a) shows the sweep lines hitting $q_1 = p_1$, $q_2 = p_2$, $q_3 = p_6$, and $q_4 = p_4$ and thus defining Q. (2b) illustrates how the q_i are used to define R. In (2c) we see that $R \subseteq Q$.

Rectangle inclusion is trivial to program and much more numerically stable than the more general quadrilateral inclusion. Figure 2 illustrates a worked example of the preprocessing algorithm which we name *eliminate*. Program 1 is a complete Pascal implementation of *eliminate*. The procedure assumes the existence of a predefined data type *point*. The array $p[1 \ldots n]$ contains the set of input points. *Eliminate* returns m, the number of points outside of R and rearranges $p[\ldots]$ so that its first m locations contain exactly those m points. The remaining points will be in $p[m+1 \ldots n]$.

We end this section by pointing out that *eliminate* has an $O(n)$ running time since all it does is make two linear passes through the data. The first pass identifies the four vertices of Q and the second disposes of the points inside R.

2. Analysis.

Let $p_i = (x_i, y_i)$, $i = 1, \ldots, n$ be Independent Identically Distributed (I.I.D.) points such that each point is uniformly distributed in the unit square $([0,1]^2)^1$. Our goal is the analysis of the asymptotic behavior of $\mathbf{E}(S)$, $n \to \infty$ where S is the random variable defined by

$$\begin{aligned} S(p_1, p_2, \ldots, p_n) &= \textit{The number of points undeleted after running the algorithm on } p_1, p_2, \ldots, p_n \\ &= \textit{The number of points outside } \mathtt{R}\ . \end{aligned}$$

Although *eliminate* is a simple, almost trivial, algorithm the analysis of $\mathbf{E}(S)$ is complicated by the fact that there is extreme probabilistic dependence between the values of q_k and even more so between the different boundaries of R. We will overcome this problem by finding the expectation of another simpler random variable and using it to upper bound $\mathbf{E}(S)$. To accomplish this we need to define the the following functions and regions (figure 3a):

$$\begin{aligned} \alpha_1 &= \min_{1\le i\le n} x_i + y_i & A_1(\alpha) &= \{(x,y):\ 0 \le x + y < \alpha\} \\ \alpha_2 &= 1 + \min_{1\le i\le n} x_i - y_i & A_2(\alpha) &= \{(x,y):\ -1 \le x - y \le -(1-\alpha)\} \\ \alpha_3 &= 1 - \max_{1\le i\le n} x_i - y_i & A_3(\alpha) &= \{(x,y):\ 1 - \alpha < x - y \le 1\} \\ \alpha_4 &= 2 - \max_{1\le i\le n} x_i + y_i & A_4(\alpha) &= \{(x,y):\ 2 - \alpha < x + y \le 2\} \end{aligned}$$

[1] Henceforth we will refer to the n points as being $\mathbf{U}[0,1]^2$.

```
function eliminate(N : integer) : integer;
    var         i1, i2, i3, i4, j, M : integer;
                a1, a2, a3, a4, highx, lowx, highy, lowy : real;
                t : point;
begin
    i1:=1     i2:=1;     i3:=1;     i4:= 1;
    a1:=p[1].x + p[1].y;      a3:=a1;
    a2:=p[1].x - p[1].y;              a4:=a2;
    for j:=2 to N do
        begin
        if p[j].x + p[j].y < a1 then
            begin    i1:=j; a1:=p[j].x + p[j].y;     end
        else if p[j].x +p [j].y > a4 then
            begin    i4:=j; a4:=p[j].x + p[j].y;     end;
        if p[j].x - p[j].y < a2 then
            begin    i2:=j; a2:=p[j].x - p[j].y;     end
        else if p[j].x - p[j].y > a3 then
            begin    i3:=j; a3:=p[j].x - p[j].y;     end;
        end;
    lowx:= MAX(p[i1].x, p[i2].x);      highx:= MIN(p[i3].x, p[i4].x);
    lowy:= MAX(p[i1].y, p[i3].y);      highy:= MIN(p[i2].y, p[i4].y);
    M:=0;    j:=N;
    while j>M do
        if lowx < p[j].x and p[j].x < highx
            and lowy < p[j].y and p[j].y < highy then
            j:=j-1
        else
            begin
                M := M+1;
                t := p[M];      p[M] := p[j];     p[j] := t;
            end;
    eliminate := M;
end;
```

Program 1. A full pascal implementation of *eliminate*. It is called with parameter N; the array $p[1\dots N]$ holds the points. After the function initializes the **for** loop finds the four points that minimize the functions $\pm x \pm y$. *i1* holds the index of q_1, *i2* of q_2, etc. while *a1, a2, a3, a4* hold the respective minimized values of $\pm x \pm y$. The next step calculates *lowx, highx, lowy, highy,* the boundaries of R. The **while** loop rearranges the array so that the non-eliminated points are at its bottom. When it returns the function has value M, the number of points that weren't eliminated.

These definitions have very intuitive motivations: α_k is the distance from the k-th corner to the intersection of its corresponding diagonal with the horizontal (vertical) boundary of the unit square while $A_k(\alpha)$ is the right isoceles triangle with base sides α that fits into the k-th corner of the square. As defined above α_k and A_k have the following property upon which our probabilistic analysis will strongly depend: $A_k(\alpha_k)$ is empty and its defining hypotenuse has at least the one point q_k on it .

We further define

$$X = \max(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$$
$$B_X = \{(x, y) : (x, y) \in (X, 1 - X)^2\}$$
$$Z = |\{p_i : p_i \notin B_X\}|.$$

B_X is the open square whose sides are parallel to, and a distance X away from, the sides of the unit square. Z is the number of points in B_X, the hollowed out region surrounding B_X (figure 3b). No matter

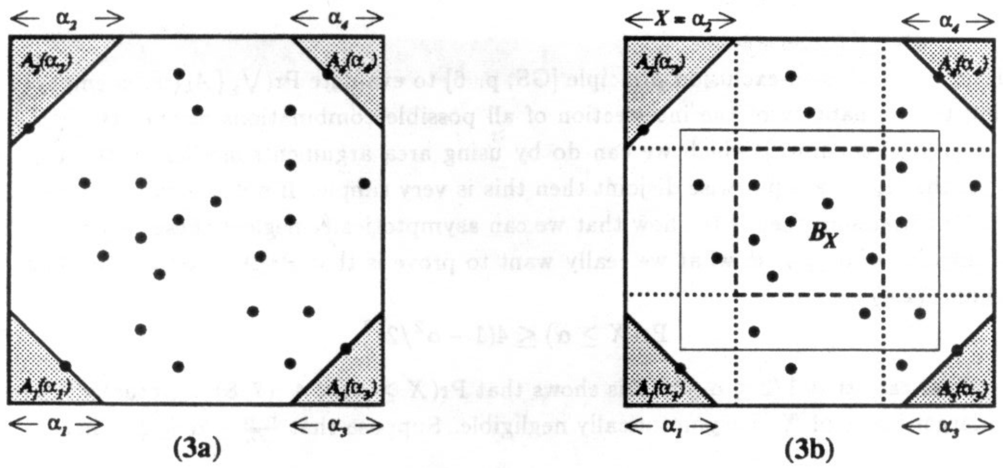

Figure 3. (3a) gives some test points with the α_k noted and the $A_k(\alpha_k)$ (shaded areas) drawn. Note especially that $A_k(\alpha_k)$ are all empty with a point (q_k) on each of their defining diagonals. (3b) is (3a) with B_X (dashed box) and $\mathtt{R}$ (thin lined box) drawn in. The dotted boxes illustrate the relationship between the α_k-s and X. The q_k are all inside the dotted boxes so $B_X \subseteq \mathtt{R}$. By inspection $Z = 17$ and $S = 10$.

what the placement of the p_i-s we always find that $\mathtt{R} \supseteq B_X$. This implies that $n - S \geq n - Z$ and thus $S \leq Z$. The objective of this section is to show that there exists a $k > 0$ such that $E(Z) \sim k\sqrt{n}$ and thus $E(S) \leq k\sqrt{n}$. Before doing this we prove the following weaker result.

Theorem 2.1: Given n points $\mathbf{U}[0,1]^2$ and X defined as above then

$$\mathrm{E}(X) \sim \frac{c}{\sqrt{n}}. \tag{2.1}$$

where $c = \sqrt{\pi}\left[\frac{7\sqrt{2}}{4} + 2\sqrt{\frac{2}{3}} - 3\right] = 1.9636\ldots$

Proof sketch: We split the proof into four parts. In the first we derive a truncated integral representation for $\mathrm{E}(X)$. In the second we derive a closed form for the integrand. In the third we sketch how to use gamma and beta function techniques to evaluate the general term of this closed form. In the fourth we pull it all together and prove the theorem.

(i) All of the α_k range between 0 (there is a point at the k-th corner) and 2 (all of the points are clustered at the corner diagonally across from the k-th one). Thus $0 \leq X \leq 2$ and we can write

$$E(X) = \int_0^2 \alpha f_X(\alpha)\,d\alpha = \int_0^2 \Pr(X \geq \alpha)\,d\alpha. \tag{2.2}$$

where $f_X(\alpha)$ is the probability density function of X. Now $X \geq \alpha$ if and only if $\exists k$ such that $\alpha_k \geq \alpha$ which in turn is true if and only if at least one of the four isoceles right triangles $A_k(\alpha)$ is empty, thus

$$X \geq \alpha \Leftrightarrow \bigvee_i \{A_k(\alpha) \text{ is empty}\} \tag{2.3}$$

($\bigvee$ indicates the union of events). If $\alpha \leq 1$ then $A_k(\alpha) \subseteq [0,1]^2$ and thus by the independence of the points

$$\Pr(A_k(\alpha) \text{ is empty}) = (1 - Area(A_k(\alpha)))^n = (1 - \alpha^2/2)^n. \tag{2.4}$$

We will use the inclusion-exclusion principle [GS, p. 6] to evaluate $\Pr(\bigvee_k\{A_k(\alpha) \text{ is empty}\})$. This will require taking the probability of the intersection of all possible combinations of one, two, three and four of the events $\{A_k(\alpha) \text{ is empty}\}$ which we can do by using area arguments similar to the type needed to derive (2.4). If the $A_k(\alpha)$ are pairwise disjoint then this is very simple. If not the calculations can get very complicated. Our first step then is to show that we can asymptotically neglect these bad cases. The $A_k(\alpha)$ are pairwise disjoint iff $\alpha \le \frac{1}{2}$ so what we really want to prove is that $\Pr(X \ge \alpha) = o(n^{-\frac{1}{2}})$ for $\alpha > 1/2$. But by (2.3) and (2.4)

$$\Pr(X \ge \alpha) \le 4(1-\alpha^2/2)^n. \tag{2.5}$$

In the special case that $1/2 \le \alpha \le 2$ this shows that $\Pr(X \ge \alpha) \le 4 \cdot (7/8)^n$. Actually 2.5 lets us show that a much larger range of X is asymptotically negligible. Suppose that $\frac{\ln n}{\sqrt{n}} \le \alpha \le 2$. Then

$$\Pr(X \ge \alpha) \le 4 \cdot (1 - \frac{\ln^2 n}{2n})^n = O(n^{-\frac{1}{2}\ln n}) \tag{2.6}$$

and therefore we can restrict the range of (2.2) to yield

$$\mathbf{E}(X) = \int_0^{\frac{\ln n}{\sqrt{n}}} \Pr(X \ge \alpha)\, d\alpha + O(n^{-\frac{1}{2}\ln n}). \tag{2.7}$$

For the sake of simplicity during the remainder of this proof we will assume that $n \ge 75$ and thus $\frac{\ln n}{\sqrt{n}} < 1/2$.

(ii) For $\alpha \le 1/2$ the $A_k(\alpha)$ are pairwise disjoint regions and thus the inclusion exclusion principle applied to (2.3) taken together with the same type of area argument used to derive (2.4) shows that

$$\Pr(X \ge \alpha) = 4(1-\alpha^2/2)^n - 6(1-2\alpha^2/2)^n + 4(1-3\alpha^2/2)^n - (1-4\alpha^2/2)^n. \tag{2.8}$$

(iii) Plugging (2.8) into (2.7) reveals that the calculation of $\mathbf{E}(Z)$ involves the evaluation of four integrals each of which has an integrand of the form $(1-k\alpha^2)^n$. This type of integral can be evaluated by transforming it into a beta function [Ru] and using Stirling's formula. Doing this we find that

$$\int_0^{\frac{\ln n}{\sqrt{n}}} (1-k\alpha^2)^n\, d\alpha \sim \frac{1}{2}\sqrt{\frac{\pi}{k}} n^{-1/2} + O(n^{-3/2}). \tag{2.9}$$

(iv) Using (2.8) and (2.9) to evaluate (2.7) we derive

$$\mathbf{E}(X) = \frac{\sqrt{\pi}}{2}\left[4\sqrt{2} - 6 + 4\sqrt{\frac{2}{3}} - \sqrt{\frac{2}{4}}\right] n^{-1/2} + O(n^{-3/2}) = cn^{-1/2} + O(n^{-3/2}) \tag{2.10}$$

where we evaluate $c = 1.9636\ldots$

Q.E.D.

The content of Theorem 2.1 is that $\mathbf{E}(X)$ is $O(n^{-1/2})$. By definition $\bar{B}_X$, the hollowed out region surrounding B_X, is composed of four strips each of average area $O(n^{-\frac{1}{2}})$ and so it also has an area of $O(n^{-1/2})$. Since there are n points this would lead us to guess that $\mathbf{E}(Z)$, the expected number of points in $\bar{B}_X$ (figure 4), is $n \cdot O(n^{-1/2}) = O(n^{1/2})$. It happens that this is true; what follows is a rigorous proof of the fact.

Theorem 2.2: Given n points $\mathbf{U}[0,1]^2$, Z and c as defined above, then

$$\mathbf{E}(Z) \sim 4c\sqrt{n}. \tag{2.11}$$

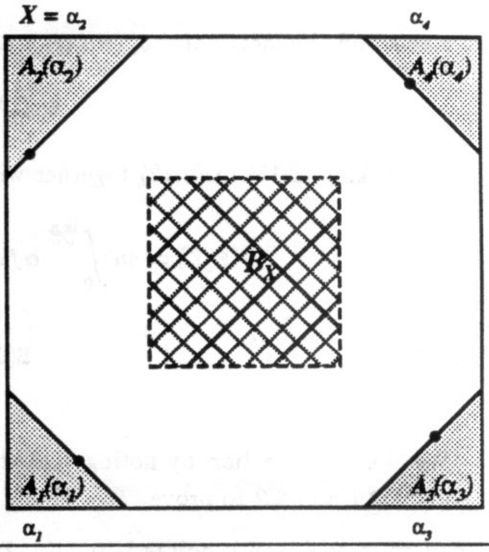

Figure 4. Fixing the α_k implies that all of the points (aside from the q_k) are independently uniformly distributed outside of $\cup_k A_k(\alpha_k)$-s (the grey areas). Thus the probability that a point is not in B_X (crosshatched region) is

$$p = \frac{\text{area of white region}}{\text{area of non gray region}} = \frac{1 - Area(B_X) - Area(\cup_k A_k(\alpha_k))}{1 - Area(\cup_k A_k(\alpha_k))}.$$

Proof sketch: We prove this in three steps. In the first step we express $\mathbf{E}(Z)$ as a truncated integral of the form $\int \mathbf{E}(Z|X=\alpha) f_X(\alpha)\,d\alpha$. In the second we derive an asymptotic expresion for $\mathbf{E}(Z|X=\alpha)$ as a function of α. In the third we prove (2.11).

(i) We start this proof in the same manner as we did the proof of Theorem 2.1. We write $\mathbf{E}(Z)$ as an integral restricted to the region $\left[0, \frac{\ln n}{\sqrt{n}}\right]$ plus an asymptotically small remainder. Notice that no matter what the value of X it is always the case that $Z \le n$. Therefore by (2.7)

$$\mathbf{E}(Z) = \int_0^{\frac{\ln n}{\sqrt{n}}} \mathbf{E}(Z|X=\alpha) f_X(\alpha)\,d\alpha + O(n^{1-\frac{1}{2}\ln n}). \tag{2.12}$$

(ii) To evaluate the conditional expectation $E(Z|X=\alpha)$ we remember that $X = \max(\alpha_1,\alpha_2,\alpha_3,\alpha_4)$ and see that we have to integrate $E(Z|\alpha_1,\alpha_2,\alpha_3,\alpha_4)$ over $\alpha_1,\alpha_2,\alpha_3,\alpha_4$ conditioned on $X=\alpha$. Formally

$$E(Z|X=\alpha) = \int E(Z|\alpha_1,\alpha_2,\alpha_3,\alpha_4) g_\alpha(\alpha_1,\alpha_2,\alpha_3,\alpha_4)\,d\alpha_1\,d\alpha_2\,d\alpha_3\,d\alpha_4 \tag{2.13}$$

where $g_\alpha(\alpha_1,\alpha_2,\alpha_3,\alpha_4)$ is the appropriate conditional probability density function for the case that $X=\alpha$.

Referring to figure 4 we see that the analysis of (2.13) is much simpler than it would first appear from its lengthy equation. The n points are uniformly distributed in $[0,1]^2$. Conditioning on $\alpha_1,\alpha_2,\alpha_3,\alpha_4$ is equivalent to conditioning on the fact that the $A_k(\alpha_k)$ are empty and that there are four points, the q_k, on their boundaries. Thus the remaining $n-4$ points are uniformly distributed in the remaining region $[0,1]^2 - \cup A_k(\alpha_k)$. This implies that each of the $n-4$ points has probability p of being in $\bar{B}_X$ where p is the probability that a point will not be in B_X conditioned on the event that $\cup A_k(\alpha_k)$ is empty so

$$p = \frac{1 - Area(B_X) - Area(\cup_k A_k(\alpha_k))}{1 - Area(\cup_k A_k(\alpha_k))} = \frac{1-(1-2\alpha)^2 - \frac{1}{2}\sum \alpha_i^2}{1-\frac{1}{2}\sum\alpha_i^2}. \tag{2.14}$$

By the independence of the points $Z|\alpha_1,\alpha_2,\alpha_3,\alpha_4$ is a random variable that has the same distribution as $4+W$ where $W(\alpha_1,\alpha_2,\alpha_3,\alpha_4)$ is a $B(n-4,p)$ (Binomial) variable. All of the α_i are less than α because $\max(\alpha_1,\alpha_2,\alpha_3,\alpha_4) = X = \alpha$ and thus, $p = 4\alpha + O(\alpha^2)$ as $\alpha \to 0$.

Since we are only interested in $\alpha \le \frac{\ln n}{n}$ we find

$$\mathbf{E}(Z|\alpha_1,\alpha_2,\alpha_3,\alpha_4) = 4 + \mathbf{E}(W) = 4n\alpha + O(\ln^2 n).$$

Using this and the fact that $\int g_\alpha(\alpha_1, \alpha_2, \alpha_3, \alpha_4)\, d\alpha_1\, d\alpha_2\, d\alpha_3\, d\alpha_4 = 1$ we integrate (2.13) and find

$$\mathbf{E}(Z|X = \alpha) = 4n\alpha + O(\ln^2 n). \tag{2.15}$$

(iii) Taking (2.15) in (2.12) together with (2.1) and (2.7) we find

$$\mathbf{E}(Z) = 4n \int_0^{\frac{\ln n}{\sqrt{n}}} \alpha f_X(\alpha)\, d\alpha + O(\ln^2 n) = 4n\mathbf{E}(X) + O(\ln^2 n)$$

and thus

$$\mathbf{E}(Z) = 4cn^{1/2} + O(\ln^2 n) \tag{2.16}$$

Q.E.D.

We close this section by noting that it is not difficult to modify the techniques used in the proofs of Theorems 2.1 and 2.2 to prove

Theorem 2.3: Given n points $\mathbf{U}[0,1]^2$, Z and c' as defined above, then

$$(a) \quad \mathrm{Var}(Z) \sim 16c'n \tag{2.17}$$
$$(b) \quad \mathbf{E}(Z^2) \sim \frac{25}{4}n \tag{2.18}$$

where $c' = \frac{25}{6} - c^2 = 0.13109\ldots$

3. Uses and Actual Performance.

In the introduction we stated that *eliminate* is meant to be used as a preprocessing step preceeding a standard convex hull algorithm. The logical question then is, for an arbitrary convex hull algorithm $\mathcal{A}$, how fast does the *eliminate*/$\mathcal{A}$ pair run compared to $\mathcal{A}$ alone? The answer obviously depends on the choice of $\mathcal{A}$. In the paragraphs that follow we'll answer this question for two standard convex hull algorithms: gift-wrapping and Graham's scan. (For a description of these algorithms and their worst and expected case running times see [PS].) Our assumptions are the same ones we've had all through this paper; the n points are $\mathbf{U}[0,1]^2$.

Theorem 3.1: The combined *eliminate*/gift-wrapping pair has an expected $O(n)$ running time.
Proof: The *eliminate*/gift-wrap pair has two stages. The *eliminate* stage is O(n) worst case and discards all but S points. When run on m points gift-wrapping has an $O(m^2)$ worst case running time. The gift-wrapping stage will therefore require at most $O(S^2) \leq O(Z^2)$ time and the combined expected running time of the pair will be $O(n) + O(\mathbf{E}(Z^2))$ which, by theorem 2.3(b), is $O(n)$.

Q.E.D.

This result is much better than the expected $O(n \log n)$ running time for the stand-alone giftwrapping algorithm with n points $\mathbf{U}[0,1]^2$. It even compares favorably with more sophisticated algorithms such as Quickhull. Quickhull also has an $O(n)$ expected and $O(n^2)$ worst case running time [OL] but has to employ a complicated nested recursion with high overhead and constants to achieve this. This is only to be expected. *Eliminate* is very much like the first step of Quickhull (eliminating some points) but we are are tailoring our choice of points to a particular distribution. Thus one elimination suffices and the algorithm is much simpler. We can extend this idea further. Quickhull runs in expected linear time over a number of distributions, ie. those where the points are uniformly distributed in a convex bounded polygon. If this distribution is known in advance then we can modify *eliminate* to work well: Find the points that are "closest" to the corners of the convex polygon (where closeness of a point is measured by the length of the projection of the line between it and the corner on the perpendicular bisector of the angle at that corner) and eliminate all of the

n	$\sqrt{n}\,\bar{X}_n$	$\bar{Z}_n/\sqrt{n}$	$\bar{S}_n/\sqrt{n}$
1000	1.874	7.025	3.410
2000	2.042	7.759	3.699
3000	1.996	7.714	3.645
4000	2.033	7.839	3.765
5000	1.849	7.235	3.386
6000	1.983	7.693	3.669
7000	1.966	7.675	3.558
8000	2.009	7.863	3.710
9000	1.966	7.696	3.645
10000	1.965	7.698	3.435

Table 1. For each value of n we chose 100 sets of n random points. For each of these sets we found X, Z, and S and then averaged over all the sets to calculate $\bar{X}_n$, $\bar{Z}_n$, and $\bar{S}_n$. Normalizing (multiplying/dividing by $\sqrt{n}$) yielded the values in the table.

points inside the convex hull of the "close" points. It is not difficult to modify our analysis of *eliminate* to show that only $O(\sqrt{n})$ points will be left on average.

For Graham's scan we achieve the following stronger result.

Theorem 3.2: The combined *eliminate*/Graham's-scan pair has an expected $O(n)$ running time. Furthermore *eliminate*'s running time strongly dominates (in an average sense) that of Graham's scan.
Proof: As in the proof of theorem 3.1 the *eliminate* stage discards all but $S \le Z$ points. When run on m points Graham's scan has an $O(m \log_2 m)$ worst case running time. Thus the expected running time of the Graham's scan is upper bounded by

$$\mathbf{E}(Z \log_2 Z) \le \mathbf{E}(Z) \log_2 n \sim 4c\sqrt{n} \log_2 n$$

which is dominated by *eliminate*'s $O(n)$ running time.

Q.E.D.

This states that the *eliminate*/Graham's-scan pair has an $O(n \log n)$ worst case time and $O(n)$ expected time, the best possible results that can be achieved by any convex hull algorithm. This is an extremely attractive result from a computational point of view especially because these bounds were achieved through an extremely simple process; The bulk of the running time is devoted to *eliminate*'s two straightfoward linear passes through the data.

In table 1 we present the results of test runs on random points. Each row contains the values $\sqrt{n}\,\bar{X}_n$ [1], $\bar{Z}_n/\sqrt{n}$, and $\bar{S}_n/\sqrt{n}$. These are the normalized values of the respective random variables averaged over 100 sets of n random points (n fixed). Theorems 2.1 and 2.3 predict that for large enough n

$$\sqrt{n}\,\mathbf{E}(X_n) \approx 1.964 \qquad \text{and} \qquad \mathbf{E}(Z_n)/\sqrt{n} \approx 7.854$$

and we do see this behavior.

4. Extensions to higher dimensions and conclusion.

[1] Until now we only implicitly noted the fact that $E(X)$, $E(Z)$, and $E(S)$ are functions of n. In order to avoid confusion in the next paragraph we will extend our notation so that X_n, Z_n, and S_n will now be the random variables for the case of n points.

It is not difficult to extend *eliminate* to d dimensions. In 2 dimensions we started by finding the four points q_1, q_2, q_3, q_4 that minimized the functions $\pm x \pm y$ (sweep lines); In d dimensions we start by finding the 2^d points $q_1, \ldots, q_{2^d}$ that minimize the functions $\pm x_1 \pm x_2 \pm \cdots \pm x_d$ (sweeping hyperplanes). We then use these points to find a rectangular d-prism $\mathtt{R}$ and eliminate all of the points inside it. The range of the i-th coordinate in $\mathtt{R}$ will be defined analogously to the x and y ranges in two dimensions. The minimum value of an ith coordinate in the range will be the maximum value of the i-th coordinate over all of the q_j that were found by hyperplanes swept in with "increasing" i-th coordinate values. Similarly the maximum value of the i-th coordinate in the range will be the minimum value over all of the q_j found by hyperplanes swept in with "decreasing" i-th coordinate values. Altough the idea is not difficult the notation involved in formal definitions is extremely cumbersome and so we will not include it. We can also extend the two dimensional analysis to the multidimensional case by redefining B_X as a specific d-cube (in place of a square) that satisfies $B_X \subseteq \mathtt{R}$. If we let S be the number of points remaining after running *eliminate* and Z the number of points outside of B_X then $S \leq Z$. Our major result, which we state without proof, is

Theorem 4.1: Given n independent identically uniformly distributed points in the unit d-cube and Z as defined above then there exists a constant $c_d > 0$ (dependent on the dimension) such that

$$\mathbf{E}(Z) \sim c_d n^{(d-1)/d}.$$

It immediately follows that $\mathbf{E}(S) \leq c_d n^{(d-1)/d}$. In three dimensions this means that $\mathbf{E}(S) = O(n^{2/3})$ and thus feeding *eliminate*'s output into an $O(n \log n)$ worst case algorithm such as Benley and Shamos' *Divide and Conquer* [BS] gives us a simple linear expected time convex hull algorithm where most of the work is done by the first step.

To review, the purpose of this paper was to prove that, given n points uniformly distributed in the unit square, *eliminate* eliminates all but $O(\sqrt{n})$ of the the original pointset without destroying any information essential to the construction of the convex hull. If the n points are distributed in the unit d-cube then *eliminate* eliminates all but $O(n^{(d-1)/d})$ of them. Therefore, following *eliminate* with an $O(n \log n)$ worst case convex hull algorithm (such as exist for 2 and 3 dimensional space) yields a pair that constructs a convex hull with a linear expected running time. Furthermore the expected running time of the actual convex hull algorithm will be sublinear. This yields a further benefit from the computational point of view because *eliminate* only performs simple comparisons (is $x > a$) while most convex hull algorithms perform more complicated geometric ones (is point P above line l). For extremely large random point sets, it is actually the case that the convex hull can be found in about the time it takes to access each point twice (once to participate in various comparisons to compute $\mathtt{R}$ and once to be eliminated).

In particular, for random point sets, the method we describe certainly outperforms "Quickhull" and is substantially easier to program. Often, the points used for the first stage of "Quickhull" are chosen to make the implementation of the recursive step more convenient: our results show that a proper choice makes the recursive step unnecessary.

It should be made clear that our choice of the unit square (d-cube) as the rectangular support of the point distribution was purely expository; The analysis of *eliminate* in section two (four) works just as well (after scaling) if the points are uniformly distributed in any rectangle (rectangular d-prism) oriented so that its sides are parallel to the Cartesian axes.

A final point that should be made is that there is another classic perspective on convex hulls which, while not yielding an analysis of *eliminate*, does provide some intuitive rationale as to why it performs as well as it does. If we are given n points uniformly identically distributed in a bounded convex polygon P then, as $n \to \infty$, the convex hull of the points will approach the perimeter of P very closely. An alternative way of expressing this is that P is a very close approximation to the convex hull. This suggests that the number of points on the hull will be comparatively small (and the number distributed inside will be relatively large). In fact this is the content of a famous theorem due to Rényi and Sulanke [RS]: If h signifies the number of points on the hull then $\mathtt{E}(h) \sim c \log n$, where c is a function of the number of edges of P.

In our case P is the unit square. Since the rectangle $\mathtt{R}$ constructed by *eliminate* is defined by the four points "closest" to the corners we can see $\mathtt{R}$ as a very close approximation of the square and therefore of the convex hull (and X can be seen as a measure of this closeness). The points that are left will are squeezed between the perimeter of the hull and the perimeter of $\mathtt{R}$ and therefore, intuitively, should be few in number.

6. REFERENCES.

[BS] J.L. Bentley and M.I. Shamos, "Divide and Conquer for Linear Expected Time," *Information Processing Letters* **7**(2) (Feb. 1978) 87-91.

[Ed] W.F. Eddy, "A new convex hull algorithm for planar sets," *ACM Transactions on Mathematical Software,* **3** (1977).

[GS] G.R. Grimmet and D.R. Stirzaker, *Probability and Random Processes,* Clarendon Press, Oxford. (1985).

[OL] M.H. Overmars and J. van Leeuwen, "Further Comments on Bykat's Convex Hull Algorithm," *Information Processing Letters,* **10**(4,5) (July 1980) 209-212.

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

[RS] A. Rényi and R. Sulanke, "Ueber die konvexe Hulle von n zufallig gewahlten Punkten, I," *Z. Wahrschien,* **2** (1963) 75-84.

[Ru] W. Rudin, *Principles of Mathematical Analysis,* 3d ed., McGraw Hill, New York. (1976).

[Se] R. Sedgewick, *Algorithms*, Addison-Wesley, Reading Mass. (1983).