Algorithmica (1996) 15: 223–241

Algorithmica

Incremental Topological Flipping Works for Regular Triangulations[1]

H. Edelsbrunner[2] and N. R. Shah[2]

Abstract. A set of n weighted points in general position in $\mathbb{R}^d$ defines a unique regular triangulation. This paper proves that if the points are added one by one, then flipping in a topological order will succeed in constructing this triangulation. If, in addition, the points are added in a random sequence and the history of the flips is used for locating the next point, then the algorithm takes expected time at most $O(n \log n + n^{\lceil d/2 \rceil})$. Under the assumption that the points and weights are independently and identically distributed, the expected running time is between proportional to and a factor $\log n$ more than the expected size of the regular triangulation. The expectation is over choosing the points and over independent coin-flips performed by the algorithm.

Key Words. Geometric algorithms, Grid generation, Regular and Delaunay triangulations, Flipping, Topological order, Point location, Incremental, Randomized.

1. Introduction. Delaunay triangulations, and their dual Voronoi diagrams, play an important role in a variety of different disciplines of science (see, e.g., the survey of Aurenhammer [2]). The computational aspects of Delaunay triangulations have been studied in the area of geometric algorithms [10], [23], and a number of different algorithms have been proposed. This paper considers the class of regular triangulations which includes the Delaunay triangulations [21]. A finite point set in $\mathbb{R}^d$ defines a unique Delaunay triangulation, but there are many regular triangulations of the set. A unique regular triangulation is implied if each point is assigned a real number as its weight. If all weights are the same, then the regular triangulation is the Delaunay triangulation of the set.

Several algorithms proposed for Delaunay triangulations are based on the notion of a local transformation, henceforth referred to as a flip. Historically, the first such algorithm is due to Lawson [18], see also [19]. Given a finite point set in the real plane, $\mathbb{R}^2$, the algorithm first constructs an arbitrary triangulation of the set. This triangulation is then gradually altered through a sequence of edge-flips until the Delaunay triangulation is obtained. The generalization of this method to $\mathbb{R}^3$ has difficulties, and Joe [16] demonstrates that it is indeed incorrect if the flips are applied to an arbitrary initial triangulation. In a different paper, Joe [17] shows that if a single point, p, is added to the

[1] The research of both authors was supported by the National Science Foundation under Grant CCR-8921421 and the research by the first author was also supported under the Alan T. Waterman award, Grant CCR-9118874. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the National Science Foundation.
[2] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA.

Received July 2, 1993; revised January 26, 1994. Communicated by B. Chazelle.

Delaunay triangulation of a set S in $\mathbb{R}^3$, then at least one sequence of flips will succeed in constructing the Delaunay triangulation of $S \cup \{p\}$. This can be used as the basis of an incremental algorithm. Rajan [25] considers Delaunay triangulations in arbitrary dimensions, $\mathbb{R}^d$, and argues that a single point can always be added by a sequence of flips. However, he needs a priority queue to find the appropriate sequence, which takes logarithmic time per flip. On a different front, Guibas et al. [15] study the complexity of the incremental algorithm in $\mathbb{R}^2$ when the points are added in a random sequence. While $\Theta(n^2)$ edge-flips are required in the worst case, they prove that under a random insertion sequence the expected number of flips is only $O(n)$. They also provide an elegant, and in the expected sense efficient, technique for locating the triangle that contains the point to be added. This step has been a sore point of all prior incremental methods.

This paper unifies and extends the algorithmic results of Joe [17], Rajan [25], and Guibas *et al*. [15]. In particular, we show that there is a sequence of flips that can be used to add a single point to a regular triangulation in $\mathbb{R}^d$. This eliminates the need for a priority queue [25] that sorts the flips. The priority queue is replaced by a stack used to generate a topological ordering of the flips in constant amortized time per flip. We use this result to generalize the incremental method of [15] to regular triangulations and to arbitrary dimensions $\mathbb{R}^d$. The resulting algorithm is similar to but not the same as the ones in [3], [6], [22]. Without any assumptions on the point distribution, the resulting algorithm runs in expected time $O(n \log n + n^{\lceil d/2 \rceil})$. The expectation is over all possible outcomes of coin-flips performed by the algorithm. The size of the regular triangulations varies widely depending on the distribution of the points and weights. Assume that the weighted points are independently and identically distributed, so that the expected number of simplices in the regular triangulation is $f(n)$. Then the expected running time of our algorithm is $O(\sum_{i=1}^{n} f(n/i))$. For example, if the points are chosen from the uniform distribution over $[0, 1]^d$ and all weights are zero, then $f(n) = \Theta(n)$ and the expected running time is $O(n \log n)$. For distributions with $f(n) = \Theta(n^{1+\varepsilon})$, $\varepsilon > 0$, we have $\sum_{i=1}^{n} f(n/i) = O(f(n))$, so the expected running time is as good as it can be.

Outline. Section 2 defines regular triangulations and introduces related terminology. Section 3 explains the relationship between regular triangulations in $\mathbb{R}^d$ and convex hulls in $\mathbb{R}^{d+1}$. This relationship provides a sometimes helpful alternative view of all concepts and techniques discussed in this paper, see also [8]. Section 4 discusses the anatomy of flipping in $\mathbb{R}^d$. A counterexample to a nonincremental method that attempts to construct regular triangulations in $\mathbb{R}^2$ by flipping is presented in Section 5. A minimalist data structure for storing triangulations is described in Section 6. The incremental algorithm is given in Section 7, and its correctness is proved in Section 8. The analysis of the algorithm given in Section 9 follows the example of [6].

2. Regular Triangulations

Triangulations. We begin by defining the notion of a triangulation used in this paper. For $0 \le k \le d$, the convex hull of a set T of $k + 1$ affinely independent points is a *k-simplex*, denoted by σ_T. The simplices σ_U, $U \subseteq T$, are the *faces* of σ_T. A collection of simplices, $\mathcal{K}$, is a *simplicial complex* if:

(i) The faces of every simplex in $\mathcal{K}$ are also in $\mathcal{K}$.
(ii) If $\sigma_T, \sigma_{T'} \in \mathcal{K}$, then $\sigma_T \cap \sigma_{T'} = \sigma_{T \cap T'}$.

Condition (ii) implies that the intersection of any two simplices in the complex is a possible empty face of both; condition (i) implies that it also belongs to the complex. The *underlying space* of $\mathcal{K}$ is the pointwise union of its simplices. Let S be a finite point set in $\mathbb{R}^d$. Usually, a triangulation of S is defined as a simplicial complex so that S is the set of 0-simplices (vertices) and the underlying space of the complex is the convex hull of S. It is convenient to relax the first condition: a simplicial complex $\mathcal{K}$ is a *triangulation* of S if:

(i) Each vertex of $\mathcal{K}$ is a point in S.
(ii) The underlying space of $\mathcal{K}$ is $\mathrm{conv}(S)$.

Notice that the second condition implies that all extreme points of S are vertices of every triangulation of S.

Power Distance and Power Diagrams. Again, let S be a finite set of points in $\mathbb{R}^d$, and assign a real valued weight w_p to each point $p \in S$. For each p, define $\pi_p\colon \mathbb{R}^d \to \mathbb{R}$ so that

$$\pi_p(x) = |xp|^2 - w_p,$$

where $|xp|$ is the Euclidean distance between points $x = (x_1, x_2, \ldots, x_d)$ and $p = (p_1, p_2, \ldots, p_d)$. $\pi_p(x)$ is usually referred to as the *power distance* of x from p. It is easy to see that for points $p, q \in S$, the locus of points $x \in \mathbb{R}^d$ with $\pi_p(x) = \pi_q(x)$ is the hyperplane

$$\chi_{p,q}\colon 2\sum_{i=1}^{d} x_i(q_i - p_i) + \sum_{i=1}^{d}(p_i^2 - q_i^2) - w_p + w_q = 0.$$

We call $\chi_{p,q}$ the *chordale* of the weighted points p and q.

Sometimes it is convenient to interpret a point p with weight w_p as a sphere $(p, \sqrt{w_p})$ with center p and radius $\sqrt{w_p}$. If $w_p \geq 0$ and x lies outside the sphere thus defined, then $\pi_p(x)$ is the square of the length of a tangent line segment from x to the sphere. $\pi_p(x)$ is also called the power of x with respect to the sphere $(p, \sqrt{w_p})$. If w_p is negative, then the sphere has an imaginary radius.

Let $H_{p,q}$ denote the half-space of points $x \in \mathbb{R}^d$ for which $\pi_p(x) \leq \pi_q(x)$. For each $p \in S$, define its *power cell* as

$$P_p = \bigcap_{q \in S - \{p\}} H_{p,q}.$$

Observe that P_p is a possibly empty convex polyhedron, the intersection of the interiors of any two distinct power cells is empty, and the union of all power cells P_p, $p \in S$, covers $\mathbb{R}^d$. The collection of power cells defines the *power diagram* $\mathcal{P}(S)$ of S, see, e.g., [1].

Orthogonal Centers. For the remainder of the paper, we assume that the weighted points of S are in general position. This involves no loss of generality since we can use the method in [13] to simulate this assumption computationally. General position, in this context, means that for every $d + 1$ weighted points in S, there is a unique unweighted point $x \in \mathbb{R}^d$ with the same power distance from all $d + 1$ points, and for every $d + 2$ weighted points of S, there is no such point. Two weighted points p and z are said to be *orthogonal* if

$$|pz|^2 = w_p + w_z,$$

that is, when the spheres $(p, \sqrt{w_p})$ and $(z, \sqrt{w_z})$ are orthogonal. Note that this is equivalent to $\pi_p(z) = w_z$ and $\pi_z(p) = w_p$. A subset T of $d+1$ (weighted) points of S defines a unique d-simplex $\sigma = \sigma_T = \text{conv}(T)$. There is a unique weighted point $z = z_\sigma$ that is orthogonal to all weighted points $p \in T$. We call z the *orthogonal center* of σ. If the weights of all $p \in T$ are zero, then the sphere with center z and radius $\sqrt{w_z}$ is the circumsphere of σ.

Local and Global Regularity. Observe that $\pi_z(p) = w_p$ for all $p \in T$. Call σ (*globally*) *regular* if $\pi_z(q) > w_q$ for all $q \in S - T$. Clearly, if σ is regular, then z is a vertex of $\mathcal{P}(S)$, the power diagram of S. The regular d-simplices, together with their faces, define a simplicial complex known as the *regular triangulation* of S, denoted by $\mathcal{R}(S)$. At this point, it is not clear that $\mathcal{R}(S)$ is a simplicial complex; this is shown in Section 3. There is a close relationship between the regular triangulation and the power diagram of S. Indeed, $\mathcal{R}(S)$ is a geometric realization of the nerve of the set of power cells, that is, $\sigma_T \in \mathcal{R}(S)$ iff $\bigcap_{p \in T} P_p \neq \emptyset$. If the weights of all points in S are zero, then $\mathcal{P}(S) = \mathcal{V}(S)$, the Voronoi diagram of S [28], and $\mathcal{R}(S) = \mathcal{D}(S)$, the Delaunay triangulation of S [7].

It is possible that the power cell of a point $p \in S$ is empty. In this case p is not a vertex of $\mathcal{R}(S)$ and we refer to p as a *redundant* point. In general, the vertex set of $\mathcal{R}(S)$ is a subset of S, namely the set of nonredundant points of S. If p is a vertex of conv(S), then $P_p \neq \emptyset$, so p is necessarily nonredundant. This implies that the underlying space of $\mathcal{R}(S)$ is indeed the convex hull of S, as required by our definition of a triangulation.

Consider an arbitrary triangulation $\mathcal{T}$ of S. Let $\sigma = \sigma_U$ be a $(d-1)$-simplex of $\mathcal{T}$ incident to d-simplices $\sigma' = \sigma_{U \cup \{a\}}$ and $\sigma'' = \sigma_{U \cup \{b\}}$. Let $z' = z_{\sigma'}$ be the orthogonal center of σ'. Then σ is said to be *locally regular* in $\mathcal{T}$ if $w_b < \pi_{z'}(b)$; otherwise, it is *locally nonregular*. Notice that if σ is locally regular in $\mathcal{T}$, then this does not imply that it belongs to $\mathcal{R}(S)$. Still, we have the following lemma which is an extension of a lemma for Delaunay triangulations proved in [7]. This lemma can also be obtained as a straightforward corollary to Lemma 3.1.

LEMMA 2.1. *If the vertex set of $\mathcal{T}$ contains all nonredundant points of S and all $(d-1)$-simplices of $\mathcal{T}$ are locally regular, then $\mathcal{T} = \mathcal{R}(S)$.*

The Power Increases. The proof of Lemma 2.1 can be based on a property of regular triangulations expressed in Lemma 2.2 below. Let σ' and σ'' be two d-simplices of a regular triangulation that share a common $(d-1)$-simplex $\sigma = \sigma' \cap \sigma''$. Let $z' = z_{\sigma'}$ and $z'' = z_{\sigma''}$ be their orthogonal centers. For every vertex v of σ, $\pi_{z'}(v) = \pi_{z''}(v) = w_v$. Hence the chordale $\chi_{z',z''}$ is the hyperplane that contains σ. So if p is a point on the same side of $\chi_{z',z''}$ as σ', then $\pi_{z'}(p) < \pi_{z''}(p)$. This implies the following lemma which is important in Section 8 when we prove the correctness of the algorithm in Section 7. The lemma can also be found in [11].

LEMMA 2.2. *Consider a half-line emanating from a point* p, *let* $\sigma_1, \sigma_2, \ldots, \sigma_{k+1}$ *be the* d*-simplices of a regular triangulation that intersect the half-line in this sequence, and define* $z_i = z_{\sigma_i}$. *Then* $\pi_{z_i}(p) < \pi_{z_i+1}(p)$, *for* $1 \le i \le k$.

3. Lifting Regular Triangulations. This section reviews the relationship between regular triangulations in $\mathbb{R}^d$ and convex hulls in $\mathbb{R}^{d+1}$. For a point $p = (p_1, p_2, \ldots, p_d) \in \mathbb{R}^d$ with weight $w_p \in \mathbb{R}$, define its *lifted point*

$$p^+ = (p_1, p_2, \ldots, p_d, p_{d+1}) \in \mathbb{R}^{d+1},$$

where $p_{d+1} = \sum_{i=1}^d p_i^2 - w_p$. For a set $S \subseteq \mathbb{R}^d$, define $S^+ = \{p^+ \mid p \in S\}$. Let (g, g^{-1}) be a pair of functions defining a *polar map*, where g maps a nonvertical hyperplane h: $x_{d+1} = 2\sum_{i=1}^d a_i x_i + a_{d+1}$ in $\mathbb{R}^{d+1}$ to the point

$$g(h) = (a_1, a_2, \ldots, a_d, -a_{d+1}) \in \mathbb{R}^{d+1},$$

and g^{-1} maps a point $p \in \mathbb{R}^{d+1}$ to the hyperplane $g^{-1}(p)$ so that $g(g^{-1}(p)) = p$.

Lemma 3.1 below expresses the close relationship between regular triangulations in $\mathbb{R}^d$ and convex hulls in $\mathbb{R}^{d+1}$. It is based on the embedding of $\mathbb{R}^d$ as the d-dimensional subspace $x_{d+1} = 0$ in $\mathbb{R}^{d+1}$. A facet of a convex polytope in $\mathbb{R}^{d+1}$ is a *lower facet* if the hyperplane that contains it is nonvertical and the polytope lies vertically above this hyperplane. By this we mean that the point $(0, \ldots, 0, +\infty)$ and the polytope lie on the same side of the hyperplane.

The lemma follows from the following fact, which is easily checked algebraically. Consider two weighted points p and z in $\mathbb{R}^d$, and let h be the hyperplane $g^{-1}(z^+)$. It is nonvertical by definition and can thus be viewed as a function from $\mathbb{R}^d$ to $\mathbb{R}$. Let $h(p)$ be its function value at $p \in \mathbb{R}^d$ and recall that $p_{d+1} = \sum_{i=1}^d p_i^2 - w_p$ is the $(d+1)$st coordinate of p^+. Then

$$\pi_z(p) - w_p = p_{d+1} - h(p).$$

This implies that $w_p > \pi_z(p)$ iff p^+ lies vertically below h, $w_p = \pi_z(p)$ iff $p^+ \in h$, and $w_p < \pi_z(p)$ iff p^+ lies vertically above h.

LEMMA 3.1. *Let* S *be a finite set of weighted points in* $\mathbb{R}^d$. *The vertical projection of the lower facets of* $\mathrm{conv}(S^+)$ *into* $\mathbb{R}^d$ *gives the* d*-simplices of* $\mathcal{R}(S)$.

We can now reinterpret Lemmas 2.1 and 2.2 in the light of Lemma 3.1. A locally regular $(d-1)$-simplex of a triangulation corresponds to a locally convex ridge (that is, $(d-1)$-face) of the polytope whose lower facets project to the d-simplices of the triangulation. Redundant points in $\mathbb{R}^d$ correspond to lifted points that are not vertices of lower facets, and the problem of constructing a regular triangulation becomes one of constructing a convex hull. Indeed, there are similarities between the work in this paper and earlier work on convex hull algorithms. Noteworthy examples are the algorithms of [6], [27]. In the second paper, Seidel discusses a convex hull algorithm based on the notion of a line-shelling of its faces, see [4]. Computing the line-shelling is closely related to computing the ordering of the flips for Delaunay triangulations mentioned in [25]. The result of this paper can also be interpreted as finding a "topological line-shelling."

4. Flipping in d Dimensions

Definition and Classification of Flips. Consider a set T of $d+2$ points in $\mathbb{R}^d$. According to Lawson [20], there are exactly two ways to triangulate T. Indeed, the two ways correspond to the two sides (lower and upper) of the $(d+1)$-simplex that is the convex hull of the corresponding lifted points in $\mathbb{R}^{d+1}$. Because of Radon's theorem (Theorem 4.1 below) and because the $(d+1)$-simplex exhausts all $d+2$ d-simplices as facets, there can be no other triangulation of T. A *flip* is the operation that substitutes one triangulation of T for the other.

In $\mathbb{R}^2$ we distinguish two cases depending on whether the tetrahedron of the lifted points in $\mathbb{R}^3$ projects to a triangle or a quadrilateral, see Figure 4.1. A 4-simplex in $\mathbb{R}^4$ projects to a single or a double tetrahedron (the convex hull of four or five points) in $\mathbb{R}^3$. Flips in $\mathbb{R}^3$ are classified accordingly, see Figure 4.2.

Given $d+2$ weighted points in $\mathbb{R}^d$, one of the two triangulations is the regular triangulation of the points, the other is not regular. In the construction of $\mathcal{R}(S)$, flips are applied in this directional sense, substituting the regular triangulation of $d+2$ points for the nonregular one.

Flippability. Let $\sigma = \sigma_U$ be a $(d-1)$-simplex of an arbitrary triangulation $\mathcal{T}$ of S, and let $\sigma' = \sigma_{U \cup \{a\}}$ and $\sigma'' = \sigma_{U \cup \{b\}}$ be the two incident d-simplices, assuming they exist. The *induced subcomplex* of $T = U \cup \{a, b\}$ consists of all simplices in $\mathcal{T}$ spanned by

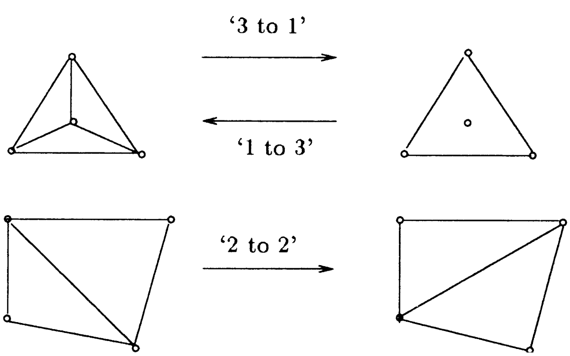

Fig. 4.1. There are three types of flips in $\mathbb{R}^2$, and we denote a flip by the number of triangles before and after the flip. So the flips are of type "1 to 3," "2 to 2," and "3 to 1." The first type introduces a new point, and the last type removes a point. The last type of flip is not needed for Delaunay triangulations because no point is redundant, and so no point has to be removed from the triangulation.

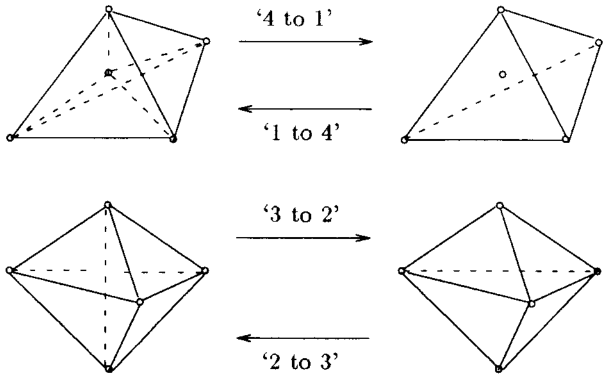

Fig. 4.2. The flips in $\mathbb{R}^3$ can be classified as "4 to 1," "3 to 2," "2 to 3," and "1 to 4."

points in T. Clearly, σ, σ', and σ'' belong to the induced subcomplex of T. We call T (and σ) *flippable* if $\text{conv}(T)$ is the underlying space of the induced subcomplex of T.

Assume that if σ is given, then σ' and σ'', and therefore T, can be computed in constant time. This requires that d be a constant. Consider the d $(d-2)$-simplices of σ. Call such a $(d-2)$-simplex *convex* if there is a hyperplane that contains it and σ' and σ'' lie on the same side of this hyperplane; otherwise, call the $(d-2)$-simplex *reflex*. The underlying space of the induced subcomplex of T is equal to $\text{conv}(T)$ iff all reflex $(d-2)$-simplices of σ have degree 3 in the induced subcomplex of T, that is, each is incident to exactly three $(d-2)$-simplices of the induced subcomplex of T. Thus the σ_V of σ are incident to three d-simplices σ', σ'', and $\sigma_{V\cup\{a,b\}}$ which fill the reflex wedge at σ_V. Thus, given σ, it is possible to test in constant time whether or not it is flippable. Recall, however, that our algorithm would attempt to flip σ only if it is flippable *and* it is locally nonregular.

A Convex Geometry Theorem. The above discussion is closely related to a classical result in convex geometry known as Radon's theorem [24]:

THEOREM 4.1. *Let T be a set of $d+2$ points in $\mathbb{R}^d$. Then a partition $T = U \dot{\cup} V$ exists so that* $\text{conv}(U) \cap \text{conv}(V) \neq \emptyset$.

Every subset of $d+1$ points of T either contains U as a subset, or V, but not both. In the former case the d-simplex spanned by the subset belongs to the triangulation of T that contains $\text{conv}(U)$ as a simplex. In the latter case it belongs to the other triangulation that contains $\text{conv}(V)$.

Assume that $|U| \le |V|$. Then $|U| \le (d+2)/2$ and σ_U is a k-simplex with $k = |U|-1 \le d/2$. When T is flipped then σ_U can only belong to one of the two triangulations of T, the one before or the one after the flip. This implies that, for every flip in $\mathbb{R}^d$, there is at least one simplex of dimension at most $d/2$ that is removed or introduced by the flip. This observation will be useful in Section 9 where we analyze the algorithm of Section 7.

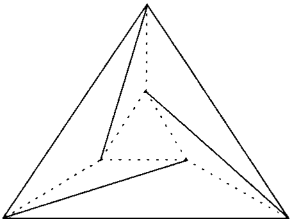

Fig. 5.1. The weights of the six points are chosen so that the solid edges are locally regular while the dotted edges are not. This is achieved if the lifted points of the three boundary vertices lie in a horizontal plane h and the lifted points of the other three vertices lie in a horizontal plane h' above h.

5. Counterexample to Nonincremental Flipping. As mentioned in the Introduction, Lawson's algorithm constructs Delaunay triangulations in $\mathbb{R}^2$ by flips that are applied to an initial triangulation. More specifically, the algorithm starts with an arbitrary triangulation of the point set and performs a sequence of "2 to 2" flips until the triangulation is the Delaunay triangulation. Joe [16] demonstrates that the generalization of this strategy to $\mathbb{R}^3$ does not always correctly compute the Delaunay triangulation. The reason is that there can be locally nonregular triangles that are not flippable. Such triangles can also appear when flips are applied after adding a single point to a Delaunay or regular triangulation (Section 7). However, in contrast to the nonincremental method, we can show that in the incremental case there is always some flip that can be applied (Section 8).

The purpose of this section is to show that in two dimensions the nonincremental method no longer works if instead of Delaunay triangulations we construct regular triangulations. Consider the example shown in Figure 5.1. The two-dimensional triangulation is a view of the inside of the so-called Schönhardt polytope in $\mathbb{R}^3$ [26]. This polytope is nonconvex and the smallest example of a polytope that cannot be decomposed into tetrahedra unless points other than its vertices are used as vertices. In the triangulation of Figure 5.1 all flippable edges are locally regular and all locally nonregular edges are nonflippable. Furthermore, all vertices have degree 4. Even though the triangulation is not the regular triangulation of the six weighted points, there is no flip that can change a locally nonregular configuration to a regular one. An iterative improvement by flipping is thus not possible.

6. A Minimalist Data Structure. In order to be reasonably specific when we discuss the algorithm in Section 7, we need to say a few words about how we store a triangulation in $\mathbb{R}^d$. This section describes a data structure that represents a triangulation by storing its d-simplices and their adjacencies. We make an effort to keep the data structure as simple as possible. It is specified in PASCAL-like formalism.

```
type  coordinate    =  integer;
      vertex        =  array [1..d + 1] of coordinate;
      vertex_index  =  1..n;
      simplex       =  record vertices: array [1..d + 1] of vertex_index;
                              neighbors: array [1..d + 1] of ↑ simplex
                       end.
```

The weighted input points are stored in an array whose elements are of type *vertex*. A d-simplex σ is represented by a record of type *simplex* so that σ.vertices [ ] gives the indices of its vertices. σ.neighbors[i] points to the d-simplex that shares all vertices except for σ.vertices[i]. Simplices of dimension $k < d$ are implicit in the data structure.

Let $\sigma' = \sigma_{U \cup \{a\}}$ and $\sigma'' = \sigma_{U \cup \{b\}}$ be two d-simplices incident to a $(d-1)$-simplex $\sigma = \sigma_U$. In order to decide whether or not σ is locally regular, we need to test whether w_b is smaller or larger than $\pi_{z'}(b)$, where $z' = z_{\sigma'}$. This is the same as deciding whether the point $b^+ \in \mathbb{R}^{d+1}$ lies above or below the hyperplane $g^{-1}(z'^+)$. This is the unique hyperplane that contains the lifted vertices of σ' (see Section 3). Such a test is described in [13]. σ is flippable iff all its reflex $(d-2)$-simplices have degree 3. Let σ_V be a $(d-2)$-simplex of σ, and let $\{c\} = U - V$. σ_V is reflex iff b and c lie on different sides of the hyperplane through $V \cup \{a\}$. Assuming σ_V is reflex, it has degree 3 iff σ_T, with $T = V \cup \{a, b\}$, is a d-simplex of the triangulation. σ_T, if present, is incident to the $(d-1)$-simplices $\sigma_{V \cup \{a\}}$ and $\sigma_{V \cup \{b\}}$. We conclude that with the above data structure, the local regularity and flippability of σ can be tested in constant time. Similarly, a flip can be carried out in constant time because the total number of d-simplices involved, the ones deleted and the ones added, is only $d + 2$.

7. The Algorithm. The algorithm constructs the regular triangulation of a given set $S = \{p_1, p_2, \ldots, p_n\}$ of weighted points incrementally, that is, points are added one at a time. It is convenient first to construct an artificial d-simplex, $\sigma_{S_0} = \operatorname{conv}(S_0)$, with $S_0 = \{p_{-d}, \ldots, p_0\}$, so that S is contained in it. We should also require that every d-simplex of $\mathcal{R}(S)$ is also a d-simplex of the regular triangulation of $S \cup S_0$. The $d + 1$ artificial points can be convieniently chosen at infinity. For example, set $w_{p_i} = 0$, and

$$p_{ij} = \begin{cases} 0 & \text{if} \quad -i > j, \\ +\zeta & \text{if} \quad -i = j, \\ -\zeta & \text{if} \quad -i < j, \end{cases}$$

where p_{ij} denotes the jth coordinate of p_i, for $-d \le i \le 0$. The symbol "ζ" is a place-holder for a large enough number, and this is the easiest way to think of the artificial points and their effect on the computations. The particular choice of points guarantees that $\mathcal{R}(S)$ is a subcomplex of $\mathcal{R}(S_0 \cup S)$. In fact, $\mathcal{R}(S)$ consists of all simplices of $\mathcal{R}(S_0 \cup S)$ that are not incident to any point of S_0.

Global Algorithm. Define $S_i = \{p_{-d}, p_{-d+1}, \ldots, p_i\}$. We proceed as follows. Given $\mathcal{R}(S_{i-1})$, let $\sigma = \sigma_T$ be the d-simplex that contains p_i. If, even after adding p_i, σ is still regular, then $\mathcal{R}(S_i) = \mathcal{R}(S_{i-1})$. Otherwise, flip $T \cup \{p_i\}$. This is a flip of type "1 to $d + 1$." Continue flipping locally nonregular $(d-1)$-simplices until none remain. The resulting triangulation is $\mathcal{R}(S_i)$.

We need some more terminology. A $(d-1)$-simplex σ_U of a triangulation belongs to the *link* of vertex p_i if $\sigma_{U \cup \{p_i\}}$ is a d-simplex of the triangulation. The $(d-1)$-simplices of the link of p_i are called *link facets*. In the algorithm given below only locally nonregular link facets are flipped.

1 Construct $\mathcal{R}(S_0) = \sigma_{S_0}$;
2 **for** $i := 1$ **to** n **do**
3 locate the d-simplex σ_T in $\mathcal{R}(S_{i-1})$ that contains p_i;
4 **if** $\mathcal{R}(T \cup \{p_i\}) \neq \sigma_T$ **then**
5 flip $T \cup \{p_i\}$;
6 **while** there are locally nonregular link facets **do**
7 find a locally nonregular link facet σ that is flippable;
8 flip σ
9 **endwhile**
10 **endif**
11 **endfor**.

In Section 8 we argue that it is indeed sufficient to restrict our attention to link facets when we search for a remaining nonregular $(d-1)$-simplex in step 7. The details of the **while** loop (steps 6–9) and the point-location operation (step 3) are explained below. As we will see, the implementation of steps 3 and 4 is slightly different than indicated above, that is, sometimes p_i is discarded even before σ_T is found.

Finding and Flipping Link Facets. We now describe a way to implement steps 6 and 7 efficiently. A stack of link facets is maintained. Each time a link facet σ is flipped, all new link facets are pushed onto the stack. The search for a link facet that is locally nonregular and also flippable begins at the top of the stack. If the topmost link facet is not flippable or it is locally regular or it is not part of the current triangulation, then it is simply popped from the stack. In the first case it could be that this link facet becomes flippable later as the result of some changes in its neighborhood. If this happens, then a neighboring link facet will be added whose flip implies the flip of the popped link facet. Consider the case where the link facet σ is no longer in the current triangulation. σ is stored in the stack as a pair of pointers to the two d-simplices incident to it. Both d-simplices are no longer part of the current triangulation. To handle this case, the d-simplices removed by flips are marked. If the two d-simplices incident to a link facet are marked, it is discarded. In fact, the d-simplices destroyed by flips are maintained in a structure called the *history dag*, see below. Each flip adds at most d facets to the stack. This implies that the total time required by the **while** loop is proportional to the number of flips performed.

Point Location. The method we use to implement step 3 is a generalization of the two-dimensional technique of [15]. The history of performed flips is used as an aid in the search. More specifically, as points are added and flips are carried out, we maintain the collection of discarded d-simplices in a directed acyclic graph, called the *history dag*.

The history dag has a unique root, which is the d-simplex σ_{S_0}. At any moment, the d-simplices of the current triangulation are the sinks of the dag. Recall that a flip replaces some k d-simplices of the current triangulation with some other $d+2-k$ d-simplices. Before the flip, the k d-simplices are sinks of the dag. Performing the flip means adding the $d+2-k$ new d-simplices as successors to the k old d-simplices. Thus, the k sinks become inner nodes, and $d+2-k$ new sinks are added to the dag.

The search with a point p_i proceeds as follows. Starting at the root of the history dag, we follow the path of d-simplices that contain p_i. Before proceeding from a d-simplex

σ_T to the next one, we check whether $w_{p_i} < \pi_z(p_i)$, where $z = z_{\sigma_T}$. If it is, then the search terminates because this implies that p_i is redundant in $T \cup \{p_i\} \subseteq S_i$ and therefore also in S.

8. Correctness. The algorithm of Section 7 could fail for two reasons. First, if all link facets are locally regular although there are other locally nonregular $(d-1)$-simplices, then the algorithm would stop before reaching the regular triangulation. We show this cannot happen. Second, it could be that the **while** loop does not terminate, either because it cycles in an infinite loop of flips or none of the locally nonregular link facets is flippable. Again we show this is impossible. We begin with a basic property of regular triangulations.

Regular d-Simplices Maximize Power Distance. Consider a subset T of $d+1$ weighted points of S, define $\sigma = \sigma_T$, and let y be a point in the interior of σ. Define $z = z_\sigma$, $h_\sigma = g^{-1}(z^+)$, and $f_y(\sigma) = \pi_z(y) = |zy|^2 - w_z$. Since h_σ is a nonvertical hyperplane we can think of it as a function from $\mathbb{R}^d$ to $\mathbb{R}$ and write $h_\sigma(y)$ for its function value at $y = (y_1, y_2, \ldots, y_d) \in \mathbb{R}^d$. As observed in Section 3, we have

$$f_y(\sigma) = y_{d+1} - h_\sigma(y),$$

where $y_{d+1} = \sum_{i=1}^{d} y_i^2$. In order to maximize f_y, over all d-simplices σ defined by S that contain y, we therefore need to minimize $h_\sigma(y)$. So h_σ must be the hyperplane spanning the lower facet of $\operatorname{conv}(S^+)$ that intersects the vertical line through y. This lower facet is the lifted version of the d-simplex σ of $\mathcal{R}(S)$ that contains y. We thus have the following result which has been proved in [25] for d-dimensional Delaunay triangulations.

LEMMA 8.1. *Let S be a finite set of weighted points in $\mathbb{R}^d$, and let $y \in \operatorname{conv}(S)$ be a point that does not lie on the hyperplane spanned by any d points of S. Over all d-simplices σ defined by S that contain y, $f_y(\sigma)$ is maximized iff σ is in $\mathcal{R}(S)$.*

For example, consider the case where a set T of $d+2$ points plays the role of S in the above lemma. Assume that T is flippable within some triangulation, and let y be a point in the interior of the convex hull of T. Let σ' and σ'' be the d-simplices that contain y, where σ' belongs to the triangulation of T before the flip and σ'' to the one after the flip. Then $f_y(\sigma') < f_y(\sigma'')$ because the triangulation after the flip is the regular triangulation of T and σ'' is part of it.

Link Facets Suffice. We show that all flips applied in the course of adding the next point, p_i, satisfy the following two properties. Let T be the set of $d+2$ points that is flipped.

(i) $p_i \in T$.
(ii) $\sigma_{\text{old}} = \sigma_{T-\{p_i\}}$ is a d-simplex in $\mathcal{R}(S_{i-1})$, and the flip destroys it.

Properties (i) and (ii) certainly hold for the first flip, performed in step 5, which adds p_i with a "1 to $d+1$" flip. Assume inductively that (i) and (ii) hold for the first $j-1$

flips. This implies that all d-simplices generated by these $j-1$ flips have p_i as a vertex. So all d-simplices of the current triangulation that are disjoint from p_i are d-simplices of $\mathcal{R}(S_{i-1})$. This implies that if $T \subseteq S_{i-1}$ is a flippable configuration, then it is locally regular and T would not be flipped. Thus, property (i) also holds for the jth flip. Let T be the $d+2$ points of the jth flip and consider $U = T - \{p_i\}$ and σ_U. This d-simplex belongs to the triangulation of T either before or after the flip. To get a contradiction, assume that σ_U belongs to the triangulation after the flip, that is, σ_U is created by the flip. Take a point y in the interior of σ_U and let σ be the d-simplex in $\mathcal{R}(S_{i-1})$ that contains y. We have $f_y(\sigma_U) > f_y(\sigma)$ because the flip increases f_y and earlier flips either also increase it or leave it unchanged. However, this contradicts Lemma 8.1 which asserts that among the d-simplices spanned by S_{i-1} that contain y—and both σ_U and σ belong to this collection—f_y is maximized by σ. So we conclude that σ_U is destroyed by the jth flip, rather than created. Property (ii) follows for the jth flip and thus holds in general.

We thus proved that each flip destroys a unique d-simplex, σ_{old}, of $\mathcal{R}(S_{i-1})$. All other d-simplices destroyed by a flip share p_i as a vertex. Except in the first flip (step 5) there is at least one such d-simplex, σ'. σ_{old} and σ' share a $(d-1)$-simplex which is thus a link facet. Right before the flip happens this link facet is flippable and locally nonregular by assumption. We thus have proved that it is sufficient to restrict our attention to link facets when locally nonregular $(d-1)$-simplices are sought.

The **while** *Loop Terminates.* Notice first that the flip of T increases f_y for all points y in the interior of conv(T). For all other points $y \in \mathbb{R}^d$, f_y remains unchanged. The increase in f value can be viewed as an indication of the progress made by the algorithm. This implies that once a k-simplex is destroyed it can never be reintroduced in the future. Thus, we can be sure that the **while** loop does not get caught in an infinite loop of flips.

Finally, we show that if there are locally nonregular link facets, then at least one of them is flippable. Consider a triangulation $\mathcal{T}$ reached at some point in time during the insertion of point p_i. The d-simplices of $\mathcal{T}$ that do not belong to $\mathcal{R}(S_{i-1})$ are exactly the ones that have p_i as one of their vertices. The union of these d-simplices is a star-shaped polytope, denoted by star(p_i). The facets of star(p_i) are exactly the link facets. Let L be the collection of d-simplices σ in $\mathcal{T}$ that lie outside star(p_i) and share a link facet with star(p_i). Let L' be the subset of d-simplices in L that are incident to locally nonregular link facets. By assumption, $L' \neq \emptyset$. For each $\sigma \in L$ consider $f(\sigma) = f_{p_i}(\sigma)$ and let $\sigma_{\min} = \sigma_U$ be the d-simplex in L' that minimizes f. We prove below that $T = U \cup \{p_i\}$ is flippable.

By choice, $f(\sigma_{\min}) \leq f(\sigma)$ for all $\sigma \in L'$. All $\sigma \in L - L'$ are incident to locally regular link facets. Therefore, $w_{p_i} < \pi_z(p_i)$, where $z = z_\sigma$. This implies

$$f(\sigma_{\min}) < w_{p_i} < \pi_z(p_i) = f(\sigma).$$

In other words, $\sigma_{\min}$ minimizes f over all $\sigma \in L$. Consider a half-line, r, emanating from p_i that intersects a link facet in its relative interior. Before intersecting any other d-simplex outside star(p_i), r intersects d-simplices in L. By Lemma 2.2 f increases along the sequence of d-simplices intersecting r. Thus, if r intersects $\sigma_{\min}$, then it cannot intersect any other d-simplex outside star(p_i) before $\sigma_{\min}$. This implies that the subcomplex induced by $T = U \cup \{p_i\}$ has underlying space equal to conv(T).

In other words, T is flippable.

9. Randomized Analysis. If the points of S are added in a random sequence we can show that the expected running time of our algorithm is $O(n \log n + n^{\lceil d/2 \rceil})$ for any point distribution. Furthermore, if the points are independently and identically distributed and $f(n)$ is the expected number of simplices in $\mathcal{R}(S)$, then we can show a running time of $O(\sum_{i=1}^{n} f(n/i))$. We begin with a brief worst-case analysis of the number of flips performed.

Maximum Number of Flips. The $d+2$ points involved in a flip define $d+2$ d-simplices, each occurring either in the triangulation of the $d+2$ points before the flip or the one after the flip. So one of the two triangulations has $k \geq (d+2)/2$ d-simplices. These k d-simplices intersect in a $(d-k+1)$-simplex, with $d-k+1 \leq d/2$. Set $\mu = \lfloor d/2 \rfloor$. This implies that each flip deletes at least one μ-simplex or adds at least one. As mentioned in Section 8, a simplex is added and deleted at most once, so the number of flips cannot exceed the total number of μ-simplices defined by n points. A μ-simplex is spanned by $\mu+1$ points, so n points span $\binom{n}{\mu+1}$ μ-simplices. Note, however, that we add $d+1$ artificial points at infinity; hence the number of μ-simplices is $\binom{n+d+1}{\mu+1}$. It follows that the maximum number of flips needed for a regular triangulation of n points in $\mathbb{R}^d$ is at most $2\binom{n+d+1}{\mu+1} = O(n^{\lceil (d+1)/2 \rceil})$. This is therefore an upper bound on the worst-case storage requirement. An additional factor n appears in the worst-case running time. The rather pessimistic worst-case analysis is due to the point-location strategy and can be improved using linear programming. The randomized analysis shows, however, that such a modification is neither necessary nor appropriate. Compare with Lemma 9.2 below.

The analysis of the running time under the assumption of a random input sequence requires some additional definitions.

Terminology and k-Set Bounds. Consider an arbitrary subset T of $d+1$ points of S and let $\sigma = \sigma_T$ be the simplex defined by T. Let $z = z_\sigma$ be the orthogonal center of σ, and define

$$\Gamma_\sigma = \{p \in S \mid \pi_z(p) < w_p\}.$$

Note that $\Gamma_\sigma \cap T = \emptyset$, and that $\Gamma_\sigma = \emptyset$ iff σ is a d-simplex of $\mathcal{R}(S)$. Call $\gamma_\sigma = |\Gamma_\sigma|$ the *width* of σ.

The analysis is based on bounds for the number of d-simplices with a fixed width k. It is also necessary to consider d-simplices incident to points of S_0. For each subset $\Omega \subseteq S_0$ and for each $0 \leq k \leq n$, write G_k^Ω for the collection of subsets $T \subseteq S_n$, $|T| = d+1$, for which $T \cap S_0 = \Omega$ and $\gamma_{\sigma_T} = k$. To avoid any confusion: the k points counted by γ_{σ_T} are points in S, because the definition of Γ_σ is such that it necessarily excludes points of S_0. Furthermore, define $G_{\leq j}^\Omega = \bigcup_{k=0}^{j} G_k^\Omega$.

For nonempty Ω, the sets G_k^Ω are somewhat more natural if we consider the lifted set $S_n^+ = \{p^+ \in \mathbb{R}^{d+1} \mid p \in S_n\}$. As explained in Section 3, the orthogonal center of $\sigma = \sigma_T$, $T \subseteq S_n$, $|T| = d+1$, corresponds to the hyperplane that contains the points of T^+. The constraint that a hyperplane contain a point with some arbitrarily large or arbitrarily small coordinates (symbolized by $+\infty$ or $-\infty$) really means the hyperplane must contain a certain direction. Recall that $T \cap S_0 = \Omega$ and that T^+ contains $\omega = |\Omega|$ points with arbitrarily large or small coordinates. So the hyperplane spanned by T^+ must contain ω directions. These constraints can be expressed using the linear hull of Ω^+, $\mathrm{lin}(\Omega^+)$, which is an ω-dimensional linear subspace of $\mathbb{R}^{d+1}$. Let F_Ω be the l-dimensional linear subspace orthogonal to $\mathrm{lin}(\Omega^+)$, where $l = d + 1 - \omega$. F_Ω can be viewed as an embedding of $\mathbb{R}^l$ in $\mathbb{R}^{d+1}$.

The maximum cardinalities of the sets G_k^Ω relate to the maximum number of k-sets of a collection of points in $\mathbb{R}^l$. A k-set of a finite point set $A \subseteq \mathbb{R}^l$ is a subset $B \subseteq A$ of size k for which there is a half-space H in $\mathbb{R}^l$ with $B = A \cap H$. Write $g_k^{(l)}(A)$ for the number of k-sets of A and define $g_{\le j}^{(l)}(A) = \sum_{k=1}^{j} g_k^{(l)}(A)$. The results on k-sets that are most relevant to our analysis are both taken from [6]. Let n be the number of points in A.

$$g_{\le j}^{(l)}(A) = O(j^{\lceil l/2 \rceil} n^{\lfloor l/2 \rfloor}), \tag{1}$$

and

$$E[g_{\le j}^{(l)}(A)] = O\left(j^l f\left(\frac{n}{j}\right)\right). \tag{2}$$

Result (2) assumes that the points are independently and identically distributed and $f(n)$ is the expected number of facets of the convex hull of the n points so chosen. The proof of (1) and (2) assumes that l is a constant and j is asymptotically less than n. If j is proportional to n, then the bounds (1) and (2) are trivial. Alternatively, this bound can be obtained by a straightforward extension of the relevant calculations in [15].

The connection between the sets G_k^Ω and the concept of a k-set is based on the lifting map explained in Section 3. Consider a set $T \in G_k^\Omega$. So $|T| = d + 1$, $T \cap S_0 = \Omega$, and for $\sigma = \sigma_T$ we have $\gamma_\sigma = |\Gamma_\sigma| = k$. Let h_σ be the hyperplane in $\mathbb{R}^{d+1}$ spanned by the points in T^+. The property of the lifting map discussed immediately before Lemma 3.1 implies that $\Gamma_\sigma^+ = S^+ \cap H$ for one of the two open half-spaces H bounded by h_σ. Thus, Γ_σ^+ is a k-set of S^+. Furthermore, if $\Omega \neq \emptyset$ and $\Omega \neq S_0$, then there is an l-dimensional linear subspace, F_Ω, with $1 \le l = d + 1 - |\Omega| \le d$, orthogonal to $\mathrm{lin}(\Omega^+)$. For a point $p \in S$, let p_Ω be the orthogonal projection of p^+ into F_Ω. Extend this definition to sets, so that, for example, $S_\Omega = \{p_\Omega \mid p \in S\}$. With these definitions, $(\Gamma_\sigma)_\Omega$ is a k-set of S_Ω. So we can use the above bound on the number of k-sets and obtain the result formulated in Lemma 9.1(i).

In order to get a similar result for independently and identically distributed points, observe that the expected number of facets of the convex hull of n points is also at most $f(n)$ in every projection into an affine space with fewer dimensions. Note that this bound tends to be less accurate as the dimension l becomes smaller.

LEMMA 9.1.

(i) *For all $\Omega \subseteq S_0$, we have $|G_{\le j}^\Omega| \le g_{\le j}^{(l)}(S_\Omega) = O(j^{\lceil l/2 \rceil} n^{\lfloor l/2 \rfloor})$, where $l = d+1-|\Omega|$.*
(ii) *If the points are independently and identically distributed and the expected number of facets of their convex hull is $f(n)$, then $E[|G_{\le j}^\Omega|] \le O(j^l f(n/j))$.*

Expected Number of d-Simplices. Using Lemma 9.1, we now derive a bound for the expected number of d-simplices that appear during the construction of $\mathcal{R}(S_n)$. We also need the following observation. Consider a set $T \in G_k^{\Omega}$. The probability that $\sigma = \sigma_T$ is a d-simplex of a regular triangulation $\mathcal{R}(S_i)$, for some $1 \leq i \leq n$, is $l!/((k+1) \cdot (k+2) \cdot \cdots \cdot (k+l))$, where $l = d+1-|\Omega|$ as usual. This is because the probability is the same as the one of adding the l points in $T \cap S$ before any of the k points in Γ_σ. Call such a d-simplex σ *nontransient*.

LEMMA 9.2. *The expected number of nontransient d-simplices is $O(n^{\lceil d/2 \rceil})$ without any assumption on the distribution of the weighted points, and it is $O(n^\varepsilon + f(n))$, $\varepsilon > 0$, with the assumption of an independent and identical distribution for the weighted points such that the expected number of simplices of the regular triangulation of n such weighted points is given by a function $f(n)$ with $f(n)/n^\varepsilon$ monotonically increasing.*

PROOF. We express the expected number of nontransient d-simplices, E, in terms of probabilities. Here the expectation is solely over the outcomes of coin-flips occurring when the point-intersection order is decided. The effect of a distribution of the weighted points will enter the proof later.

$$E = \sum_{\Omega \subseteq S_0} \sum_{k=0}^{n} \sum_{T \in G_k^{\Omega}} \text{Prob}[\sigma_T \text{is nontransient}].$$

We can replace the last sum by the cardinality of G_k^{Ω} times the probability calculated above.

$$\begin{aligned} E &= \sum_{\Omega \subseteq S_0} l! \sum_{k=0}^{n} \frac{|G_k^{\Omega}|}{(k+1)(k+2) \cdot \cdots \cdot (k+l)} \\ &= \sum_{\Omega \subseteq S_0} l! \left(\sum_{k=0}^{n} \frac{|G_{\leq k}^{\Omega}|}{(k+1) \cdot \cdots \cdot (k+l)} - \sum_{k=0}^{n-1} \frac{|G_{\leq k}^{\Omega}|}{(k+2) \cdot \cdots \cdot (k+l+1)} \right) \\ &= \sum_{\Omega \subseteq S_0} l! \left(\frac{|G_{\leq n}^{\Omega}|}{(n+1) \cdot \cdots \cdot (n+l)} + \sum_{k=0}^{n-1} \frac{l \cdot |G_{\leq k}^{\Omega}|}{(k+1) \cdot \cdots \cdot (k+l+1)} \right). \end{aligned}$$

The first term in the sum over sets Ω can be neglected because

$$|G_{\leq n}^{\Omega}| \leq \binom{n+l}{l}$$

which implies that it is smaller than 1. If we now use Lemma 9.1(i), we obtain

$$E \leq \sum_{\Omega \subseteq S_0} l(l!) \sum_{k=1}^{n} \frac{cn^{\lfloor l/2 \rfloor}}{k^{1+\lfloor l/2 \rfloor}},$$

where c is some positive constant. Note that $l \leq d+1$ and $|S_0| = d+1$ are both constants because d is a constant. This implies $E = O(n^{\lceil d/2 \rceil})$, which is the first part of the assertion. If we use Lemma 9.1(ii) instead of (i), we get

$$E \leq \sum_{\Omega \subseteq S_0} l(l!) \sum_{k=1}^{n} \frac{c \cdot f(n/k)}{k} = O(f(n))$$

if we assume that $f(n) = \Omega(n^\varepsilon)$, $\varepsilon > 0$, since $f(n)/n^\varepsilon$ is monotonically increasing. This gives the second result of the lemma. $\square$

The above argument only counts nontransient d-simplices that occur during the construction of $\mathcal{R}(S_n)$. There are also transient d-simplices that occur. A *transient* d-simplex is constructed during the insertion of new point, say p_i, and is destroyed before the regular triangulation of S_i is completed. As mentioned earlier, each flip destroys one d-simplex of $\mathcal{R}(S_{i-1})$, and it creates at most $d+1$ d-simplices. It thus follows that the total number of transient d-simplices constructed by the algorithm is at most of the same order of magnitude as the number of nontransient ones. Thus, the bound in Lemma 9.2 applies also to the expected number of transient d-simplices. If we ignore the input, the amount of memory required by the algorithm in Section 7 is bounded by the size of the history dag, which is proportional to the total number of transient and nontransient d-simplices. Thus, Lemma 9.2 gives a bound on the expected memory requirement.

Point Location. The amount of time spent for locating p_i (step 3) is proportional to the length of the traversed path. The accounting is done differently for transient and for nontransient d-simplices. If σ is a nontransient d-simplex on the path of p_i, but not the last d-simplex on this path, then $p_i \in \Gamma_\sigma$. If σ is transient, then we find a nontransient d-simplex σ' with $p_i \in \Gamma_{\sigma'}$ that is not used yet in the accounting of the cost for p_i. Since σ is transient, there is a flip that removed σ from the triangulation, and this flip also removed one d-simplex of $\mathcal{R}(S_{i-1})$. This d-simplex is nontransient and we let σ' be this d-simplex. Notice that σ' is counted only once for point p_i. In summary, the point-location cost for p_i is bounded by one plus the number of nontransient d-simplices that contain p_i in their sets Γ. Therefore, n plus the sum of γ_σ over all nontransient d-simplices σ is an upper bound for the total cost that occurs in step 3 of the algorithm.

LEMMA 9.3. *The expected cost of point location is $O(n \log n + n^{\lceil d/2 \rceil})$, without any assumption on the distribution of the points, and it is $O(\sum_{k=1}^{n} f(n/k))$, with the assumption that the weighted points are independently and identically distributed and the expected number of simplices in the regular triangulation of n such weighted points is $f(n)$.*

PROOF. As in the proof of Lemma 9.2, we can compute the expectation, E, of $\sum \gamma_\sigma$ by summing probabilities. We sum over all nontransient d-simplices σ. If we use Lemma 9.1(i), we get

$$E = \sum_{\Omega \subseteq S_0} \sum_{k=0}^{n} \sum_{T \in G_k^\Omega} k \cdot \text{Prob}[\sigma_T \text{is nontransient}]$$

$$\leq \sum_{\Omega \subseteq S_0} l(l!) \sum_{k=1}^{n} \frac{cn^{\lfloor l/2 \rfloor}}{k^{\lfloor l/2 \rfloor}} = O(n \log n + n^{\lceil d/2 \rceil}),$$

where c is some positive constant and $l = d + 1 - |\Omega|$, as in Lemma 9.2. This proves the first part of the assertion. If we use Lemma 9.1(ii), we get

$$E \leq \sum_{\Omega \subseteq S_0} l(l!) \sum_{k=1}^{n} c \cdot f\left(\frac{n}{k}\right) = O\left(\sum_{k=1}^{n} f\left(\frac{n}{k}\right)\right),$$

which is the second part of the assertion. □

Apart from the point-location cost, the algorithm takes only constant time per flip. Note that the sum expressing the running time is bounded from above by $O(f(n))$ if $f(n) = \Omega(n^{1+\varepsilon})$, $\varepsilon > 0$, and $f(n)/n^{1+\varepsilon}$ monotonically increases. This shows that the algorithm runs in expected time proportional to the expected size of what it produces, unless this expected size is linear or only slightly superlinear.

An interesting special case is when the points are independently and uniformly distributed in the unit hypercube in $\mathbb{R}^d$. For the case of zero weights (in which case the regular triangulation is the Delaunay triangulation of the points), Dwyer [9] proved that $f(n) = \Theta(n)$, assuming d is a constant, see also [5]. Theorem 9.4 implies that in this case the expected running time of our algorithm is $O(n \log n)$.

We can thus summarize the results of this section, and indeed of this paper.

THEOREM 9.4. *The expected running time and memory requirement of the randomized incremental version of the algorithm in Section* 7 *are, respectively,* $O(n \log n + n^{\lceil d/2 \rceil})$ *and* $O(n^{\lceil d/2 \rceil})$. *If we assume that the weighted points are independently and identically distributed and* $f(n)$ *is the expected number of simplices in the regular triangulation of* n *such weighted points, then the expected running time is* $O(\sum_{k=1}^{n} f(n/k))$. *Furthermore, if* $f(n)/n^{\varepsilon}$ *is monotonically increasing, then the expected memory requirement is* $O(n^{\varepsilon} + f(n))$, *for any* $\varepsilon > 0$.

10. Concluding Remarks. Delaunay triangulations, and more generally regular triangulations, have a fair number of applications, including the generation of grids for point configurations and the construction of so-called alpha shapes [12], [14]. Indeed, the main motivation for studying the problems solved in this paper is our intention to implement weighted and unweighted alpha shapes in dimensions beyond $\mathbb{R}^3$. It would be interesting to conduct an experimental study comparing the algorithm of this paper with its main contenders for constructing d-dimensional regular triangulations. These are probably the randomized algorithm of Clarkson and Shor [6] and the output-sensitive algorithm of Seidel [27]. The difference between the algorithms in this paper and in [6] are in the details which nevertheless can affect their performance. It should be pointed out that Seidel's algorithm is neither randomized nor on-line. The algorithm in this paper is sensitive to the expected output size when the weighted points are independently and identically distributed.

Acknowledgments. We thank Raimund Seidel for sharing an observation that significantly simplified an earlier version of the correctness proof in Section 8. We thank Edgar

Ramos for pointing out that an earlier version of our analysis could be adapted to the case of weighted points that are independently and identically distributed. Independent from us, Barry Joe found a proof that incremental topological flipping works for computing Delaunay triangulations in d-dimensional space. His proof is similar to ours, although it covers only the case of equal weights for all points.

References

[1] F. Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM J. Comput.*, **16** (1987), 78–96.

[2] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surveys*, **23** (1991), 345–406.

[3] J.-D. Boissonnat and M. Teillaud. On the randomized construction of the Delaunay tree. *Theoret. Comput. Sci.*, **112** (1993), 339–354.

[4] H. Bruggesser and P. Mani. Shellable decompositions of cells and spheres. *Math. Scand.*, **29** (1971), 197–205.

[5] C. Buchta, J. Müller, and R. F. Tichy. Stochastical approximation of convex bodies. *Math. Ann.*, **271** (1985), 225–235.

[6] K. Clarkson and P. Shor. Applications of random sampling in computational geometry. *Discrete Comput. Geom.*, **4** (1989), 387–421.

[7] B. N. Delaunay. Sur la sphère vide. *Izv. Akad. Nauk SSSR Otdel. Mat. Est. Nauk*, **7** (1934), 793–800.

[8] O. Devillers, S. Meiser, and M. Teillaud. The space of spheres, a geometric tool to unify duality results on Voronoi diagrams. *Proc.* 4*th Canad. Conf. on Computational Geometry*, 1992, pp. 263–268.

[9] R. A. Dwyer. Higher-dimensional Voronoi diagrams in linear expected time. *Discrete Comput. Geom.*, **6** (1991), 343–367.

[10] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Heidelberg, 1987.

[11] H. Edelsbrunner. An acyclicity theorem for cell complexes in d dimensions. *Combinatorica*, **10** (1990), 251–260.

[12] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. Inform. Theory*, **29** (1983), 551–559.

[13] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graphics*, **9** (1990), 66–104.

[14] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. Manuscript, Dept. Comput. Sci., Univ. Illinois at Urbana-Champaign, 1992.

[15] L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, **7** (1992), 381–413.

[16] B. Joe. Three-dimensional triangulations from local transformations. *SIAM J. Sci. Statist. Comput.*, **10** (1989), 718–741.

[17] B. Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Comput. Aided Geom. Design*, **8** (1991), 123–142.

[18] C. L. Lawson. Generation of a triangular grid with applications to contour plotting. Memo 299, Jet Propulsion Laboratory, Pasadena, CA, 1972.

[19] C. L. Lawson. Software for C^1 surface interpolation. In *Mathematical Software III*, edited by J. Rice. Academic Press, New York, 1977, pp. 161–194.

[20] C. L. Lawson. Properties of n-dimensional triangulations. *Comput. Aided Geom. Design*, **3** (1986), 231–246.

[21] C. Lee. Regular triangulations of convex polytopes. *Applied Geometry and Discrete Mathematics*: *The Victor Klee Festschrift*, edited by P. Gritzmann and B. Sturmfels, American Mathematical Society, Providence, RI, 1991, 443–456.

[22] K. Melhorn, S. Meiser, and C. Ó'Dúnlaing. On the construction of abstract Voronoi diagrams. *Discrete Comput. Geom.*, **6** (1991), 211–224.

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

[24] J. Radon. Mengen konvexer Körper, die einen gemeinsamen Punkt enthalten. *Math. Ann.*, **83** (1921), 113–115.

[25] V. T. Rajan. Optimality of the Delaunay triangulations in $\Re^d$. *Proc.* 7*th Ann. Symp. on Computational Geometry*, 1991, pp. 357–363.

[26] E. Schönhardt. Über die Zerlegung von Dreieckspolyedern in Tetraeder. *Math. Ann.*, **98** (1928), 309–312.

[27] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. *Proc.* 18*th Ann. ACM Symp. on Theory of Computing*, 1986, pp 403–413.

[28] G. F. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *J. Reine Angew. Math.*, **133** (1907), 97–178.