INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center Street • Suite 600 • Berkeley, California 94704 • 1-510-642-4274 • FAX 1-510-643-7684

Building Convex Space Partitions Induced by Pairwise Interior-Disjoint Simplices

Marco Pellegrini*

Dept. of Computer Science, King's College London

TR-93-039

August 1993

Abstract

Given a set S of n pairwise interior-disjoint $(d-1)$-simplices in d-space, for $d \geq 3$, a Convex Space Partition induced by S (denoted $CSP(S)$) is a partition of d-space into convex cells such that the interior of each cell does not intersect the interior of any simplex in S. In this paper it is shown that a $CSP(S)$ of size $O(n^{d-1})$ can be computed deterministically in time $O(n^{d-1})$. These bounds are worst case optimal for $d = 3$. The results are proved using a variation of the efficient hierarchical cuttings of Chazelle.

*Department of Computer Science, King's College, Strand, London WC2R 2LS U.K. e-mail: marco@dcs.kcl.ac.uk

1 Introduction

1.1 Convex Space Partitions

Decomposing subsets of Euclidean Real Space E^d into convex parts is an ubiquitous problem in Computational Geometry. Most of the known results are for $d = 2, 3$ and fewer for higher dimensional spaces (i.e. $d \geq 4$). We refer the reader to [Cha87] for a survey of results in the plane. In this paper we deal with $d \geq 3$. The problem in its general formulation is the following. We are given a subset $S \subset E^d$ and we consider the free space $F_S = E^d/S$. In how many convex sets can we partition F_S? How fast can we compute such a partition? Here we examine the case when S is a set of pairwise interior-disjoint $(d-1)$-simplices. This class of input includes boundaries of simple and non-simple polyhedra in E^d. We can think of this class as an high-dimensional extension of planar maps [PS85, Ede87], which are arrangements of pairwise interior-disjoint segments. In order to solve problems on the set S, such as locate points, plan collision free paths, etc., it is important to chop the free space F_S into simple and manageable parts. Quite often these parts must be convex to facilitate further computations. Thus the problem of computing efficiently Convex Space Partitions lies at the heart of many geometric problems in 3-and higher dimensional spaces.

Aronov and Sharir in [AS90, AS92] prove that a cell C in an arrangement of n *possibly intersecting* $(d-1)$-simplices in E^d can be partitioned in $O(n^{d-1})$ convex polyhedra. Unfortunately this combinatorial result does not immediately imply an efficient algorithm to build such a partition in $d \geq 4$. For the special case $d = 3$, a convex decomposition of C of size $O(n^{2+\delta})$ can be computed in expected time $O(n^{2+\delta})$ for any given $\delta > 0$, where the constant of proportionality depends on δ. The algorithm of Aronov and Sharir is more general (for $d = 3$) than the one presented in this paper since the simplices are not required to be disjoint. On the other hand the method in [AS90, AS92], which is based on randomized techniques in [Cla87], has *expected* time and size bounds $O(n^{2+\delta})$ also for disjoint triangles. In this paper we improve on the algorithm of Aronov and Sharir, by giving a *deterministic* algorithm and by reducing the time and size bounds by an n^δ factor, when the input is composed of pairwise interior-disjoint simplices. Also, within the improved time bound we compute a decomposition of the whole free space F_S rather than a single cell.

A concept similar to that of a CSP is that of a *Binary Space Partition.* Given a set S of n interior-disjoint triangles in 3-space, a Binary Space Partition of E^3 induced by S (denoted $BSP(S)$) is a hierarchical partition of E^3 into *convex cells* associated with the nodes of a binary tree. The root is associated with the whole space E^3 and the regions associated with the children of node v form a convex partition of the region associated with v. The leaves are associated with regions whose interior does not meet any triangle in S. Paterson and Yao show in [PY90] how to obtain a BSP's of size $O(n^2)$ in time $O(n^3)$. They restrict the construction to those BSP whose cells are bounded by planes spanning triangles in S (auto-partitions). For the class of auto-partition they also show a lower bound $\Omega(n^2)$ on the number of cells of the BSP, in the worst case. The technique in [PY90] constructs a BSP of optimal worst case size but the time bound for the computation is far from optimal. In higher dimensional space the method in [PY90] builds a BSP of size $O(n^{d-1})$ in time $O(n^{d+1})$. Binary Space Partitions have applications in Constructive Solid Geometry and Hidden Surface Removal. In Constructive Solid Geometry BSP's are used to find small formulas for the description of non-convex polyhedral objects. In the context of Hidden Surface Removal BSP's are used to compute efficiently the correct visibility order of polyhedral faces with respect to a view-point (see [PY90, Mul91] for more details on applications).

Mulmuley [Mul91] uses a *cylindrical decomposition* to solve visibility problems in 3-space. The cylindrical decomposition in [Mul91], which is similar to a construction in [CEG+90], has worst case size $O(n^2)$ and is built in worst case time $O(n^2 \log n)$.[1] The method in [Mul91] is based on sweeping a plane in 3-space, together with the dynamic maintenance of a planar point location data structure. This method is off the optimal worst case time by a logarithmic factor. Moreover this sweeping approach does not give us an efficient algorithm in higher dimensional space. The main obstacle is that it is not known how to efficiently maintain convex decompositions under dynamic insertion and deletion of faces, when the decomposition to be maintained is in dimension three or greater. In this paper we take a different approach that avoids any sweeping technique and dynamic maintenance of data structures.

For the special case when S is the boundary of a simple polyhedron[2] in 3-space Chazelle [Cha84] gives a Convex Space Partition of size $O(n^2)$ which is built in time $O(n^2)$. Both bounds are worst case optimal. The method in [Cha84] relies on the connectivity properties of the boundary of a simple polyhedron and therefore this method does not extend to a set of disjoint simplices or to a non-simple polyhedron. Later, Chazelle and Palios [CP90] gave an algorithm to decompose a simple polyhedron in 3-space which produces $O(n + r^2)$ convex pieces in time $O(nr + r^2 \log r)$ where n is the number of edges of the polyhedron and r in the number of *reflex* edges. Again, it is unlikely that this technique can be extended to similar problems in higher dimensional space or to non-simple polyhedra in 3-space. Some convex space partitions for special polyhedral sets in E^3 are discussed in [Ber93].

In this paper we show how to construct Convex Space Partitions (which enclose both cylindrical decompositions and BSP's) induced by pairwise interior-disjoint $(d-1)$-simplices in d-dimensional space ($d \geq 3$). We obtain a *deterministic* algorithm that builds a Convex Space Partition of size $O(n^{d-1})$ in time $O(n^{d-1})$.

The results of this paper are obtained by modifying the the Hierarchical Cuttings Method of Chazelle [Cha93]. The main idea is to project the set S onto a $(d-1)$-subspace and construct a cutting on the projected arrangement. This computation is intermixed with additional split operations which are carried out in the original d-dimensional space.

To our knowledge no better algorithm is known for dimension $d \geq 4$ for the case when S is a set of disjoint simplices and also for the sub-case when S is the boundary of a simple polyhedron. For $d = 3$, our result improves of a logarithmic factor over the worst case time needed to construct a cylindrical decomposition using the method in [Mul91]. Moreover, the size and time bounds are worst case optimal for $d = 3$, as follows from an $\Omega(n^2)$ lower bound in [Cha84].

---

[1] Actually the time and the size of the cylindrical decomposition in [Mul91] depend on the number of "regular crossings" of the projection of the edges of S onto the xy-plane. The number of regular crossings can range from constant to quadratic in n.

[2] A simple polyhedron is a piecewise-linear 3-manifold with boundary which is homeomorphic to a closed 3-ball.

1.2 Point Location

Often in applications it is not sufficient to decompose the free space F_S in convex parts. We also need an efficient method for locating the cell containing a query point. Point location on planar maps is a very important problem and mentioning all the important papers on the subject is beyond the scope of this paper. We refer the reader to [PS85, Ede87]. For the analogous problem in dimension $d \geq 3$ fewer results are known.

If S is a set of hyperplanes then we can locate a point in time $O(\log n)$ using $O(n^d)$ storage [Cha93]. If S forms a convex partition of E^3 with n edges faces and vertices, a method of Preparata and Tamassia [PT89] locates a point in S in time $O(\log^2 n)$ using $O(n \log^2 n)$ storage. The cylindrical decomposition in [Mul91] can be associated with a point location data structure of worst case size $O(n^2)$ that answers point location queries in time $O(\log^2 n)$. Our result on CSP implies immediately a data structure of size $O(n^{d-1})$ answering in time $O(\log n)$ point location queries in F_S. Moreover, given any two query points p_1 and p_2, we can determine if they are in the same connected component of F_S and if so we give explicitly a path connecting p_1 and p_2.

The paper is organized as follows: in Section 2 we review properties of the hierarchical cuttings. In Section 3 we give the algorithm to construct the CSP.

2 Efficient Hierarchical Cuttings

In this section we give the basic definitions and lemmas used to derive the main results in the next section.

We are given a set H of n hyperplanes in general position in Euclidean d-dimensional Space E^d. Let $R \subseteq H$ be a subset of ρ hyperplanes. For a segment e let R_e (resp H_e) be the number of hyperplanes in R (resp. H) intersecting e. For a simplex s let R_s (resp H_s) be the number of vertices of the arrangement created by R (resp. H) contained in s.

Definition 1 *R is a $(1/r)$-approximation for H if, for any e:*

$$\left| \frac{R_e}{\rho} - \frac{H_e}{n} \right| < \frac{1}{r}.$$

Definition 2 *R is a $(1/r)$-net for H if for any e, $H_e > n/r$ implies $R_e > 0$.*

Definition 3 *R is a sparse $(1/r)$-net for (H,s) if for any e, $H_e > n/r$ implies $R_e > 0$; and $R_s \leq 4(\rho/n)^d H_s$.*

Definition 4 *An $(1/r)$-cutting for H is a partition of R^d into interior-disjoint simplices such that any simplex meets at most n/r of the hyperplanes in H. The number of simplices in the partition is called the size of the cutting.*

Chazelle [Cha93] builds a sequence of $(1/r_0^i)$-cuttings for H for $i = 1,..,k$, which are denoted by $C_1...,C_k$, where $k = \log_{r_0} r$, r_0 is a constant, and r a parameter $r < n$. Some properties of the hierarchical cuttings are the following:

1. Each C_i is a convex partition of R^d, where each cell of C_i is a d-dimensional simplex. By $|C_i|$ we denote the number of simplices of C_i.

2. C_i is a refinement of C_{i-1}. That is, every cell of C_i is a subset of a single cell of C_{i-1}.

3. Each cell in C_i is intersected by at most $O(n/r_0^i)$ hyperplanes in H.

4. $|C_i| \leq r_0^{(i+1)d}$.

5. The time to construct the hierarchical cutting is:

$$\sum_{1 \leq i \leq \log_{r_0} r} \left(\frac{n}{r_0^i}\right) |C_i| \leq n r_0^{k(d-1)+d} = O(n r^{d-1})$$

Let s be a simplex in R^d and H a set of n hyperplanes in R^d. We denote with $H(s)$ the subset of hyperplanes of H intersecting s, and with H_s the number of vertices of the arrangement $\mathcal{A}(H)$ within s. Let $n/r_0^{k-1} > |H(s)| > n/r_0^k$, and $\rho_0 = r_0^k |H(s)|/n$ and $\rho = \rho_0 \log \rho_0$. We use the following lemmas from [Cha93]:

Lemma 1 *Let A be an $(1/2d\rho_0)$-approximation of $H(s)$ and R a sparse $(1/2d\rho_0)$-net for (A, s), then we have $R_s \leq 4(\rho/|H(s)|)^d H_s + 4\rho^d/\rho_0$.*

Lemma 2 *A triangulation of R built as in Lemma 1 has size $O(\rho^{d-1} + R_s)$ and it is a $1/r_0$-cutting for $H(s)$ in s.*

Lemma 3 *The sparse net R of Lemma 1 is computed in time $O(|H(s)|)$.*

3 Constructing Convex Space Partitions

We fix once and for all a vertical direction in E^d.

Definition 5 *A simplex t partially covers a simplex s if t intersects s and the vertical projection of t does not completely contain the vertical projection of s.*

If t partially covers s then the vertical projection of a $(d-2)$-face of t will intersect the vertical projection of s.

Definition 6 *A simplex t completely covers a simplex s if t intersects s and the projection of t includes the projection of s.*

We are given a set S of n simplices in E^d. The construction of $CSP(S)$ proceeds in stages. We build a sequence of sets $C_1, .., C_l$ where $l = \log_{r_0} n/2$ and r_0 is a suitable constant. The set C_i is a collection of triples $(s, P(s), Q(s))$ where s is an convex cell of constant size (an elementary cell) in in d-space, $P(s)$ is the subset of simplices in S *partially covering* s, and $Q(s)$ is the subset of simplices in S *fully covering* s. If $s \in C_i$ and we will have associated sets $P(s)$ and $Q(s)$ with the invariant properties that $|P(s)| \leq n/{r_0}^i$ (first invariant) and $|Q(s)| \leq nr_0/{r_0}^i$ (second invariant).

From the construction strategy we will have that C_i is a refinement of C_{i-1} and moreover the simplices in C_i partition E^d. Easily C_l contains only simplices intersecting a constant number of original simplices in S, thus we obtain a $CSP(S)$ of size $O(|C_l|)$. We assume that the two invariants hold for C_{k-1} and we show how to construct C_k.

1. Let s be an elementary cell in C_{k-1}. Since the first and second invariant hold by inductive hypothesis, s is partially covered by $n/{r_0}^{k-1}$ simplices and is fully covered by $nr_0/{r_0}^{k-1}$ simplices.

2. The covering simplices $Q(s)$ are disjoint and therefore they are linearly ordered in the vertical direction. We split $Q(s)$ into r_0 groups of $Q(s)/r_0$ simplices each by selecting every $Q(s)/r_0$-th simplex in the vertical ordering. The selected covering simplices slice s into convex cells numbered $i = 1, .., r_0$, which we denote $\sigma(s, i)$ (or σ_i whenever s is clear from the context. We also denote with $\sigma^*(s, i)$ the vertical projection of $\sigma(s, i)$.

3. From the above construction each σ_i is covered by $|Q(\sigma_i)| \leq nr_0/{r_0}^k$ simplices. Each σ_i is partially intersected by $|P(\sigma_i)| = q_i$ hyperplanes where $\sum_i q_i = |P(s)| = n/{r_0}^{k-1}$. This property is easily proved since a partially covering simplex of $P(s)$ can belong to only one cell σ_i.

4. The average σ_i has a partial cover of size $|P(s)|/r_0$ which is exactly what is needed to become a valid element of the set C_k. To make this averaging argument work in the worst case we proceed as follows. We project, independently for each cell σ_i, all the $(d-2)$-boundaries of the partially covering simplices of $P(\sigma_i)$ onto a $(d-1)$-subspace. We extend these sets into full hyperplanes, obtaining a set $H(s, i)$ of q_i hyperplanes in $(d-1)$-space. We build the sparse net of Lemmas 1, 2 and 3 in $(d-1)$-space. So we generate elementary $(d-1)$-dimensional cells. Then each such cell is extended in the vertical direction within σ_i into a d-dimensional elementary cell.

5. We choose as parameter of the sparse net construction a number ρ_{0i} such that : $q_i/\rho_{0i} = |P(s)|/r_0$ thus $\rho_{0i} = q_i r_0/|P(s)|$. The interesting property is that:

$$\sum_i \rho_{0i} = \sum_i q_i r_0/|P(s)| = r_0/|P(s)| \sum_i q_i \leq r_0|P(s)|/|P(s)| = r_0$$

   This sum of the ρ_{0i}'s is less than r_0. Let $\rho_i = \rho_{0i} \log \rho_{0i}$.

6. Let $A_i(s)$ be the number of d-dimensional cells generated at the previous step. Using the bound of Lemma 2 and 1, the number of cells obtained is:

$$A_i(s) \leq \rho_i^{d-2} + (\rho_i/q_i)^{d-1} H_{\sigma^*(s,i)}(s,i) + \rho_i^{d-1}/\rho_{0i}$$

   Let η be a cell so obtained. We have that the number of simplices partially covering η is $|P(\eta)| \leq q_i/\rho_{0i} = P(s)/r_0 = n/{r_0}^k$ (Lemma 2), thus η satisfies the first invariant of C_k. The number of simplices in $Q(\eta)$ is at most $|Q(s)|/r_0 + |P(s)|$ thus is at most $n/{r_0}^{k-1} + n/{r_0}^{k-1} = 2nr_0/{r_0}^k$.

   We take η and use the median simplex in the vertical ordering of $Q(\eta)$ to split η into two cells η_1 and η_2 which satisfy both invariants for C_k. Collecting all of these cells we have a partition of E^d which forms C_k.

This is the end of the algorithm that builds C_k starting from C_{k-1}.

3.1 Analysis of the algorithm

We now derive a recursive equation linking the size of C_k with the size of C_{k-1}.

$$|C_k| \leq 2 \sum_{s \in C_{k-1}} \sum_{\sigma(s,i)} A_i(s) = 2 \sum_{s \in C_{k-1}} \sum_{\sigma(s,i)} [\rho_i^{d-2} + (\rho_i/q_i)^{d-1} H_{\sigma^*(s,i)}(s,i) + \rho_i^{d-1}/\rho_{0i}]$$

Bounding terms of the summation separately we have: $\sum_i \rho_i^{d-2} \leq (\sum_i \rho_{0i} \log \rho_{0i})^{d-2} \leq {r_0}^{d-2} \log^{d-2} r_0$; and $\sum_i \rho_i^{d-1}/\rho_{i0} \leq ((\sum_i \rho_{0i})^{d-2})(\log^{d-1}(\sum_i \rho_{0i})) \leq {r_0}^{d-2} \log^{d-1} r_0$.

Since $\rho_i/q_i \leq {r_0}^k(\log r_0)/n$ we obtain a term $\sum_s \sum_i ({r_0}^{k(d-1)}(\log r_0)^{d-1}/n^{d-1}) H_{\sigma^*(s,i)}(s,i)$. Clearly $\sum_s \sum_i H_{\sigma^*(s,i)}(s,i) \leq n^{d-1}$. So the recursive equation becomes:

$$|C_k| \leq 2c(\log r_0)^{d-1} {r_0}^{k(d-1)} + 2c(\log r_0)^{d-1} {r_0}^{d-2} |C_{k-1}|,$$

for a constant c which is independent of r_0.

Lemma 4 *$|C_k| \leq D{r_0}^{k(d-1)}$ for a constant D, independent of k.*

Proof. We use an induction on k. For $k = 1$ we start with an $1/r_0$-cutting of the projections of the initial set of simplices S. Easily C_1 satisfies both invariants and $|C_1| \leq D{r_0}^{d-1}$ for a sufficiently large constant D independent of r_0. Inductively we assume the bound on $|C_{k-1}|$.

$$\begin{aligned} |C_k| &\leq 2c(\log r_0)^{d-1} {r_0}^{k(d-1)} + 2cD(\log r_0)^{d-1} {r_0}^{d-2} {r_0}^{(k-1)(d-1)} \leq \\ &\leq 2c(\log r_0)^{d-1} {r_0}^{k(d-1)} + (2cD(\log r_0)^{d-1}/r_0) {r_0}^{k(d-1)} \end{aligned}$$

Choosing r_0 and D large enough we can make sure that $2c(\log r_0)^{d-1} + 2cD(\log r_0)^{d-1}/r_0 \leq D$ and thus the bound is proved. ∎

The total number of cells in the sequence of sets C_k is a summation of a geometric sequence of ratio $r_0^{d-1} > 1$ so its value is proportional to the last term, which is $O(n^{d-1})$.

$$\sum_{k=1}^{l} |C_k| \leq \sum_{k=1}^{l} D r_0^{k(d-1)} \leq 2D{r_0}^{l(d-1)} = O(n^{d-1}).$$

What is the time to compute all of the C_i's? The sparse nets are computed in time linear in the number of simplices partially covering the elementary cells. The slicing hyperplanes are found by repeated applications of a selection algorithm in time $O(r_0|Q(s)|)$ for the elementary cell s. Thus we have running time:

$$\sum_{k=1}^{l}((n/{r_0}^k) + (n{r_0}^2/{r_0}^k))|C_k| \leq 2(1+{r_0}^2)Dn{r_0}^{l(d-2)} = O(nn^{d-2}) = O(n^{d-1})$$

The above discussion proves the following theorem:

Theorem 1 *Give a set S of n pairwise interior-disjoint $(d-1)$-simplices in d-space with $d \geq 3$ we can build deterministically a Convex Space Partition $CSP(S)$ of size $O(n^{d-1})$ in time $O(n^{d-1})$.*

Following the tree structure of the convex space decomposition we can locate the cells in $CSP(S)$ containing a query point p in time $O(\log n)$.

Corollary 1 *We can locate in time $O(\log n)$ the cells of $CSP(S)$ containing a query point p. Within the same query time we can determine if p is incident to any simplex in S and we can find the simplex immediately below p.*

Within the time bounds for constructing $CSP(S)$ we can build a graph representing the adjacency relation among cells in $CSP(S)$, by visiting this graph we can determine the adjacent cells therefore we can mark each cell of the last level of $CSP(S)$ with a label that indicates the unique cell in E^d/S containing that cell.

Corollary 2 *Given a set S of n interior-disjoint simplices we build a data structure of size $O(n^{d-1})$ in time $O(n^{d-1})$ such that, given a query point p, we can locate the cell in E^d/S containing p in time $O(\log n)$.*

4 Conclusions

We have shown an algorithm to build convex space partitions induced by pairwise interior disjoint simplices. The algorithm works in any fixed dimension and is optimal for $d = 3$. We apply these partitions to solve point location and path planning problems. In a companion paper [Pel93] we use similar techniques to solve point location and path planning problems in arrangements of intersecting simplices.

References

[AS90] B. Aronov and M. Sharir. Triangles in space or building (and analyzing) castles in the air. *Combinatorica*, 10(2):137–173, 1990.

[AS92] B. Aronov and M. Sharir. Castles in the air revisited. In *Proceedings of the 8th ACM Symposium on Computational Geometry*, pages 146–256, 1992.

[Ber93] M. Bern. Compatible tetrahedralizations. In *Proceedings of the 9th ACM Symposium on Computational Geometry*, pages 281–288, 1993.

[CEG+90] K.L. Clarkson, H. Edelsbrunner, L.J. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete & Computational Geometry*, 5:99–160, 1990.

[Cha84] B. Chazelle. Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM J. on Computing*, 13(3):488–507, 1984.

[Cha87] B. Chazelle. Approximation and decomposition of shapes. In *Advances in Robotics, Vol. 1: Algorithmic and Geometric Aspects of Robotics*, pages 145–185. Lawrence Erlbaum Associates, 1987.

[Cha93] B. Chazelle. Cutting hyperplanes for divide and conquer. *Discrete & Computational Geometry*, 9:145–158, 1993.

[Cla87] K.L. Clarkson. New applications of random sampling in computational geometry. *Discrete & Computational Geometry*, 2:195–222, 1987.

[CP90] B. Chazelle and L. Palios. Triangulating a nonconvex polytope. *Discrete & Computational Geometry*, 5:505–526, 1990.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer Verlag, 1987.

[Mul91] K. Mulmuley. Hidden surface removal with respect to a moving view point. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 512–522, 1991.

[Pel93] M. Pellegrini. Point Location, Vertical Ray Shooting and Motion Planning in Arrangements of Simplices. Manuscript, August 1993.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: an Introduction.* Springr Verlag, 1985.

[PT89] F. Preparata and R. Tamassia. Efficient spatial point location. In *Proceedings of the 1989 Workshop on Algorithms and Data Structures*, pages 3–11, 1989.

[PY90] M.S. Paterson and F.F. Yao. Efficient binary space partitions for hidden surface removal and solid modeling. *Discrete & Computational Geometry*, 5:485–503, 1990.