

On Range Searching with Semialgebraic Sets*

P. K. Agarwal¹ and J. Matoušek²

¹ Computer Science Department, Duke University,
Durham, NC 27706, USA

² Katedra Aplikované Matematiky, Universita Karlova,
118 00 Praha 1, Czech Republic
and

Institut für Informatik, Freie Universität Berlin,
Arnimallee 2-6, D-14195 Berlin, Germany
matoušek@cspguk11.bitnet

Abstract. Let P be a set of n points in \mathbb{R}^d (where d is a small fixed positive integer), and let Γ be a collection of subsets of \mathbb{R}^d , each of which is defined by a constant number of bounded degree polynomial inequalities. We consider the following Γ -range searching problem: Given P , build a data structure for efficient answering of queries of the form, “Given a $\gamma \in \Gamma$, count (or report) the points of P lying in γ .” Generalizing the simplex range searching techniques, we give a solution with nearly linear space and preprocessing time and with $O(n^{1-1/b+\delta})$ query time, where $d \leq b \leq 2d - 3$ and $\delta > 0$ is an arbitrarily small constant. The actual value of b is related to the problem of partitioning arrangements of algebraic surfaces into cells with a constant description complexity. We present some of the applications of Γ -range searching problem, including improved ray shooting among triangles in \mathbb{R}^3 .

1. Introduction

Let Γ be a family of subsets of the d -dimensional space \mathbb{R}^d (d is a small constant) such that each $\gamma \in \Gamma$ can be described by some fixed number of real parameters (for example, Γ can be the set of balls, or the set of all intersections of two ellipsoids,

* Part of the work by P. Agarwal was supported by National Science Foundation Grant CCR-91-06514. Part of the work by J. Matoušek was supported by a Humboldt Research Fellowship. A preliminary version of this paper appeared in *Proc. 17th Symp. on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, Vol. 629, Springer-Verlag, Berlin, 1992, pp. 1–13.

etc.; see below for a more formal definition). We consider the following algorithmic problem:

Given a set P of n points in \mathbb{R}^d , build a data structure, which answers the queries of the following form efficiently: Given a query object $\gamma \in \Gamma$, count (or report) all points of P lying in γ .

Actually, we consider a more general setting, where a weight function on points in P is assumed and the cumulative weight of points in $P \cap \gamma$ is required. The weights are assumed to belong to a semigroup, i.e., subtractions are not allowed. We assume that the semigroup operations can be executed in constant time. As is typical in computational geometry, we use the *real RAM* model of computation, where the input data contains arbitrary real numbers and each arithmetic operation with real numbers is charged unit cost. We also assume that the roots of a fixed-degree polynomial can be computed in constant time.

A special case of the Γ -range searching problem that has been intensively studied is the *simplex range searching*, where Γ is the set of all d -dimensional simplices. This simplex range searching is by now reasonably well understood: lower bounds were given by Chazelle [10], and nearly matching upper bounds were given by Chazelle *et al.* [16] and further improved by Matoušek in [17] and [29] (some of the several previous significant works on this problem include [17], [22], [34], and [35]). Ignoring various subpolynomial factors, these results essentially say that the simplex range searching problem can be solved either with linear storage and preprocessing and $O(n^{1-1/d})$ query time, or with a polylogarithmic query time and $O(n^d)$ storage. These two solutions can be combined to construct a data structure of size m , $n \leq m \leq n^d$, so that a query can be answered in time $O(n/m^{1/d})$.

There is an important range searching problem, which can be viewed as a special case of the simplex range searching problem, but which admits a more efficient solution—the *half-space emptiness* problem. Here Γ is the set of all half-spaces, and we are only interested in determining whether a query half-space contains any point of P . By the results of [19] and [28], this problem can be solved (again ignoring subpolynomial factors) with $O(n/m^{1/\lfloor d/2 \rfloor})$ query time using space m , $n \leq m \leq n^{\lfloor d/2 \rfloor}$. An extension to reporting points in the query half-space is also possible, with the number of reported points added to the query time, but no such result is known, e.g., for counting the points in a query half-space.

Only few results were published for the nonlinear case, when the objects of Γ are bounded by surfaces other than hyperplanes. One well-studied case is reporting (or counting) points in query disks in the plane [6], [17], or in query balls in higher dimensions, since it is closely related to the *nearest neighbor problem*, *k-nearest neighbors problem*, etc. Chazelle and Welzl [17] give linear-space solutions to the circular range searching problem in the plane with $O(\sqrt{n} \log^2 n)$ query time. All k points lying in a query disk can be reported in time $O(\log n + k)$ using a data structure of size $O(n \log n)$ [6], [28].

The paper by Chazelle and Welzl [17] provides an elegant general result, which bounds a certain measure of complexity of the Γ -range searching problem on a

set P in terms of the so-called Vapnik–Chervonenkis dimension of the set system $(P, \{P \cap \gamma \mid \gamma \in \Gamma\})$ (this notion is explained in Section 2). Unfortunately, this result only estimates the number of semigroup operations needed to answer the query, but does not account for other operations needed in the algorithm. Hence, their result does not immediately give an algorithm. However, in some special cases, e.g., in the case of the disk range searching, their approach can be made algorithmic by using some additional data structures (see the original paper for details).

The techniques developed for the simplex range searching seem to be quite powerful and general, and many researchers felt that they should be applicable to the general Γ -range searching problem, where ranges are defined by constant number of bounded-degree polynomial inequalities. A basic example arises as follows: Let p be a constant and let $f(x_1, \dots, x_d, a_1, \dots, a_p)$ be a fixed $(d + p)$ -variate polynomial of degree bounded by some constant.¹ Let us consider a collection Γ_f of subsets of \mathbb{R}^d defined by

$$\Gamma_f = \{\gamma_f(a) \mid a \in \mathbb{R}^p\}, \quad (1.1)$$

where

$$\gamma_f(a) = \{x \in \mathbb{R}^d \mid f(x, a) \geq 0\}$$

for some $a \in \mathbb{R}^p$.

In this paper we show that, indeed, the known techniques can be extended to handle the Γ -range searching problem, where $\Gamma = \Gamma_f$. The extension of simplex range searching to this setting is relatively straightforward, but one technical difficulty has to be overcome, namely, the construction of the so-called “guarding set” (or “test set”), where the most direct translation of the method used in the simplex case does not work.

Our algorithms easily extend to ranges defined by conjunctions and disjunctions of a bounded number of polynomial inequalities. Disjunctions correspond to unions of ranges, and by rewriting the defining formula suitably, we can assume that these are disjoint unions (e.g., a formula $A \vee B$ can be rewritten to $A \vee (B \& \text{NOT } A)$), and disjoint unions are straightforward to deal with in range searching. A conjunction of polynomial inequalities defined by the same polynomial f (i.e., $\gamma = \gamma_1 \cap \gamma_2$ such that $\gamma_1, \gamma_2 \in \Gamma_f$) can be handled without any additional effort; see Section 6 for details. On the other hand, a conjunction of inequalities defined by different polynomials is handled using *multilevel data structures*. Roughly speaking, having a suitable Γ_1 -range searching data structure and a Γ_2 -range searching data structure, the technique of multilevel data structures enables us to build a Γ -range searching data structure, where $\Gamma = \{\gamma_1 \cap \gamma_2 \mid \gamma_1 \in \Gamma_1, \gamma_2 \in \Gamma_2\}$.

A multilevel data structure can be viewed as a “composition” of the range searching data structures. Perhaps surprisingly (at the first sight), the efficiency of

¹ Throughout this paper, a_i, x_i are used to denote the coordinates of points a and x , respectively, and a^i, x^i are used to denote sequences of points.

the resulting data structure is roughly the same as the efficiency of the worst one of the data structures we started with. A simple instance of multilevel data structures is the *range tree* (see [31]). An example more closely related to our application was given by Dobkin and Edelsbrunner [20]. Chazelle *et al.* [16] show how a multilevel data structure can be used to extend a half-space range searching data structure to answer simplex range queries. Other applications of multilevel structures can be found in [27] and [5]. An abstract framework for multilevel data structures has been introduced in [29].

We can also handle strict inequalities in the same manner as nonstrict ones (choosing nonstrict inequalities for the presentation is just an arbitrary convention). Therefore, we can build range searching data structures for arbitrary *semialgebraic sets* of a *constant description complexity*. A semialgebraic set in \mathbb{R}^d is a set obtainable from a finite number of sets of the form $\{x \in \mathbb{R}^d \mid f(x) \geq 0\}$ or $\{x \in \mathbb{R}^d \mid f(x) > 0\}$ (f is a d -variate polynomial with rational coefficients) by Boolean operations (disjunction, conjunction, complementation). A semialgebraic set has a constant description complexity if it is defined by a constant number of polynomial inequalities (the constant may depend on the dimension), each of constant maximum degree; such semialgebraic sets are sometimes called *Tarski cells*.

The efficiency of our Γ_f -range searching algorithms is determined by the ability to partition the cells in arrangements of certain algebraic surfaces into small numbers of some “elementary” subcells with a constant size description. We consider a Γ_f -range searching structure with linear space. In our method we first obtain a certain “closure” $(d + \bar{p})$ -variate polynomial $\tilde{f}(x, \bar{a})$, $\bar{a} = (\bar{a}_1, \dots, \bar{a}_{\bar{p}})$, from the polynomial f . In general $\bar{p} \geq p$. The construction of this new polynomial is not canonical (that is, there may be different choices for \tilde{f}). If we can partition the arrangement of any collection of m surfaces of the form

$$\{\{x \in \mathbb{R}^d \mid \tilde{f}(x, \bar{a}^i) = 0\} \mid \bar{a}^1, \dots, \bar{a}^m \in \mathbb{R}^{\bar{p}}\}$$

into $O(m^b)$ elementary cells, then we can construct a linear-size data structure for a Γ_f -range searching with query time $O(n^{1-1/b+\delta})$.² The preprocessing time of our algorithm is $O(n \log n)$. The most natural preprocessing algorithm is randomized, but it can also be turned into a deterministic one without affecting the asymptotic running time. Our data structures achieve only a relatively small saving in the query time (a factor of $n^{1/b}$ at best) compared with the (trivial) $O(n)$ bound.

It is also possible to get a (poly)logarithmic query time, provided that we can afford a relatively large space, and we can get a tradeoff between space and query time as well. We briefly outline these possibilities here, without giving any details. In order to get a Γ_f -range searching data structure with $O(\log n)$ query time, we may generalize the method developed by Chazelle *et al.* [16] for half-spaces in a straightforward way. Essentially, it requires preprocessing the arrangement of certain algebraic surfaces for answering point-location queries. Here, again, the

² We use δ to denote an arbitrarily small positive constant. The multiplicative constants in the asymptotic bounds may depend on δ .

efficiency of the data structure (the space requirement) depends on the size of certain decompositions of arrangements of algebraic surfaces. This time the decomposition takes place in the parameter space: the relevant surfaces in \mathbb{R}^p are $\{a \in \mathbb{R}^p \mid f(x^i, a) = 0\}$ for some $x^1, \dots, x^m \in \mathbb{R}^d$. Assuming that the arrangement of any m such surfaces can be decomposed into $O(m^\beta)$ elementary cells, the Γ_f -range searching problem can be solved with $O(n^{\beta+\delta})$ space and preprocessing time, and with $O(\log n)$ query time. The point-location problem in an arrangement of algebraic surfaces is discussed in [12]. The data structures with logarithmic query time can be combined with the linear-space ones to obtain a space/query-time tradeoff; see, e.g., [16].

The paper is organized as follows. In Section 2 we give basic definitions and we discuss the so-called *linearization* of range spaces. In Section 3 we extend the notion of cuttings to an abstract framework. Section 4 discusses various methods for decomposing an arrangement of algebraic surfaces into constant size cells. Section 5 derives the main results of the paper, and Section 6 describes the range searching algorithms based on the results derived in the previous section. We present some applications of Γ -range searching, and finally conclude in Section 8 by mentioning some open problems.

2. Range Spaces and Linearization

We recall some terminology and results concerning range spaces, since they offer a convenient abstract framework for our considerations; see [22], [24], and [7] for more material.

Range Spaces. A *range space* is a pair (X, Γ) , where X is some (possibly infinite) set and Γ is a set of subsets of X . The elements of X are usually called the *points* and the elements of Γ the *ranges* of (X, Γ) . For $Y \subseteq X$, the *subspace of (X, Γ) induced by Y* is defined as the range space $(Y, \{\gamma \cap Y \mid \gamma \in \Gamma\})$ (the ranges are considered as sets). The *Vapnik–Chervonenkis dimension* or *VC-dimension* of a range space (X, Γ) is d if there is no subset $A \subseteq X$ of size $d + 1$ such that the subspace induced by A has all possible subsets of A as ranges.

Typical examples of range spaces in computational geometry are of the form (\mathbb{R}^d, Γ) , where Γ is a set of geometric figures, such as the set of all half-spaces, all simplices, all balls, or it has the form Γ_f for some polynomial f (see above), etc. In computational applications, one usually encounters subspaces of these range spaces induced by finite subsets of \mathbb{R}^d . Let \mathcal{H}_d denote the range space

$$(\mathbb{R}^d, \{\gamma \mid \gamma \text{ a half-space in } \mathbb{R}^d\}).$$

All these examples (and various others) share an important property that they have finite VC-dimension.

(1/r)-Nets. Let (X, Γ) be a range space with X finite. A subset $S \subseteq X$ is called a *(1/r)-net* for (X, Γ) , provided that $S \cap \gamma \neq \emptyset$, for every $\gamma \in \Gamma$ with $|\gamma|/|X| > 1/r$.

Haussler and Welzl [22] proved that, given a range space of VC-dimension at most d and a parameter $r > 0$, a $(1/r)$ -net of size at most $C_d r \log r$, where C_d is a constant depending only on d , exists. Recently, Komlós *et al.* [24] showed that $C_d = d + o(d)$ and that, in the worst case, this bound is tight up to an additive factor.

Embeddings and Linearizations. If $f: X \rightarrow Y$ is a mapping from a set X to a set Y and $A \subseteq X$, we let $f(A) = \{f(x) | x \in A\} \subseteq Y$. For a set $B \subseteq Y$, we define $f^{-1}(B) = \{x \in X | f(x) \in B\}$ (note that we do not assume $B \subseteq f(X)$).

Let (X, Γ) and (Y, Σ) be range spaces, and let $\varphi: X \rightarrow Y$ be an injective function. Then φ is called an *embedding* of range spaces if, for each $\gamma \in \Gamma$, there is a $\sigma \in \Sigma$ such that $\gamma = \varphi^{-1}(\sigma)$. In other words, for every $\gamma \in \Gamma$, a $\sigma \in \Sigma$ exists whose intersection with $\varphi(X)$, the image of X in Y , is precisely $\varphi(\gamma)$. We let $\varphi^*(\gamma)$ denote some σ with this property (in general, more than one such σ may exist; it will be irrelevant which one is chosen for $\varphi^*(\gamma)$ in our considerations). We call the embedding φ *faithful* if $\varphi^{-1}(\sigma) \in \Gamma$ for every $\sigma \in \Sigma$. Given an embedding φ , we can always add more subsets to Γ so that φ becomes faithful.

A basic nontrivial example of embedding of range spaces in computational geometry arises from the well-known “lifting to the paraboloid”, see, e.g., [21]. Here $X = \mathbb{R}^2$ (the plane), Γ is the set of all (closed) disks, (Y, Σ) is the space \mathcal{H}_3 , and the mapping $\varphi: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is given by $\varphi((x_1, x_2)) = (x_1, x_2, x_1^2 + x_2^2)$. The set $f(X)$ is the unit paraboloid. For a set $B \subseteq \mathbb{R}^3$, its inverse image $\varphi^{-1}(B)$ is obtained by first intersecting B with the unit paraboloid, and then vertically projecting the intersection onto the xy -plane. We claim that φ is an embedding of range spaces. Obviously it is injective, and it is well known that the image of any disk D in the plane can be obtained as the intersection of the unit paraboloid with a closed half-space h in \mathbb{R}^3 ; $\varphi^*(D) = h$. In this particular case $\varphi^*(D)$ is unique. The embedding is not a faithful one, but if we also add the complements of all open disks and the empty set to Γ , φ becomes faithful. A generalization of this example is discussed below.

We need embeddings to the range spaces \mathcal{H}_k ; we call such an embedding $\varphi: X \rightarrow \mathbb{R}^k$ of a range space (X, Γ) into the range space \mathcal{H}_k a *linearization* of (X, Γ) ; k is called the *dimension* of this linearization.

As far as we could find out, very little is known about linearizations of range spaces. Alon *et al.* [7] introduce the notion of embedding and give some negative examples. Yao and Yao [35] made an observation which, translated into our terminology, says that each range space of the form (\mathbb{R}^d, Γ_f) , f a bounded degree $(d + p)$ -variate polynomial, is linearizable: Indeed, to get a linearization in this case, it suffices to express the multivariate polynomial $f(x, a)$ in the form

$$f(x, a) = \psi_0(a) + \psi_1(a)\varphi_1(x) + \cdots + \psi_k(a)\varphi_k(x) \quad (2.1)$$

for some real functions $\varphi_1, \dots, \varphi_k, \psi_0, \dots, \psi_k$. Then a point $x \in X$ is mapped to the point

$$\varphi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_k(x)) \in \mathbb{R}^k. \quad (2.2)$$

Then for a range $\gamma_f(a) = \{x \in \mathbb{R}^d \mid f(x, a) \geq 0\}$, the corresponding range $\varphi^*(a)$ is the half-space

$$\{y \in \mathbb{R}^k \mid \psi_0(a) + \psi_1(a)y_1 + \cdots + \psi_k(a)y_k \geq 0\}. \quad (2.3)$$

Algorithm for Linearizing Polynomials. A simplest way to express the polynomial $f(x, a)$ in the form (2.1) is to write f as a sum of monomials in the x_i variables with its coefficients being polynomials in the a_j 's. Then each monomial in the x_i 's will play the role of one function φ_i , and its coefficient will be the corresponding function ψ_i (ψ_0 is formed by the terms containing no x_i 's).

It is often important to minimize the dimension of a linearization. The method just described does not necessarily give the best result. Let us return to the example with disks in the plane considered above. A circle with center (a_1, a_2) and radius a_3 in the plane can be regarded as a set of the form $\gamma_f(a)$, where $a = (a_1, a_2, a_3)$ and f is the 5-variate polynomial

$$f(x_1, x_2, a_1, a_2, a_3) = a_3^2 - (x_1 - a_1)^2 - (x_2 - a_2)^2.$$

A straightforward application of the above method yields a linearization of dimension 4. However, f can be written in the form

$$f(x_1, x_2, a_1, a_2, a_3) = [a_3^2 - a_1^2 - a_2^2] + [2a_1x_1] + [2a_2x_2] - [x_1^2 + x_2^2], \quad (2.4)$$

thus, setting

$$\begin{aligned} \psi_0(a) &= a_3^2 - a_1^2 - a_2^2, & \psi_1(a) &= 2a_1, & \psi_2(a) &= 2a_2, & \psi_3(a) &= -1, \\ \varphi_1(x) &= x_1, & \varphi_2(x) &= x_2, & \varphi_3(x) &= x_1^2 + x_2^2, \end{aligned}$$

we get a linearization of dimension 3. It corresponds to the above discussed "lifting" to the unit paraboloid.

We describe an efficient algorithm for transforming a given polynomial $f(x, a)$ into the form (2.1) with k as small as possible. We restrict ourselves to the case when each ψ_i and each φ_i is a polynomial. Let us write $f(x, a)$ in the form

$$f(x, a) = \psi_0(a) + \sum_{\mu \in M} \left(\sum_{v \in N_\mu} c_{\mu, v} a^v \right) x^\mu.$$

Here M is the set of all distinct nonzero d -component vectors appearing as exponents of the x_i -variables, N_μ is the set of exponent vectors for monomials in the a_j -variables appearing in the coefficient at x^μ , and $c_{\mu, v}$ are real coefficients. For a vector $\mu = (\mu_1, \mu_2, \dots, \mu_d)$, put $x^\mu = x_1^{\mu_1} x_2^{\mu_2} \cdots x_d^{\mu_d}$. Let $N = \bigcup_{\mu \in M} N_\mu$, and let C be the matrix

$$C = (c_{\mu, v})_{\mu \in M, v \in N}.$$

Theorem 2.1. *The minimum value of k needed to write a polynomial $f(x, a)$ in the form (2.1) with ψ_i, φ_i polynomials is equal to the rank of the matrix C .*

Proof. First we observe that we can assume each φ_i to be a linear combination of monomials x^μ , for $\mu \in M$,

$$\varphi_i = \sum_{\mu \in M} u_{i,\mu} x^\mu,$$

where $u_{i,\mu}$ is a real coefficient. Similarly each ψ_i is a linear combination of monomials a^ν , for $\nu \in N$,

$$\psi_i = \sum_{\nu \in N} v_{i,\nu} a^\nu,$$

where $v_{i,\nu}$ is a real coefficient. This follows from the linear independence of the monomials x^μ over $\mathbb{R}[a]$ for distinct μ , and the linear independence of the monomials a^ν over $\mathbb{R}[x]$ for distinct ν . In fact, in this way it can be shown that it does not help to use various other “reasonable” functions (such as \sin , \exp , \ln) in the formulas for φ_i, ψ_i , but it is not obvious that some very “wild” functions could not be used. That is why we explicitly restrict ourselves to linearizations given by polynomials.

Further, we can assume that none of the polynomials φ_i has a nonzero absolute term (since such a term can be transferred to ψ_0), so we assume that M does not contain the zero vector.

With the above notation, if the equality (2.1) should hold, we get the system of equations

$$c_{\mu,\nu} = \sum_{i=1}^k u_{i,\mu} v_{i,\nu}, \quad \mu \in M, \quad \nu \in N. \quad (2.5)$$

We want to determine the vectors $u_i = (u_{i,\mu})_{\mu \in M}$ and $v_i = (v_{i,\nu})_{\nu \in N}$. The minimum value of k needed to satisfy (2.5) is equal to the rank of the matrix C (we are indebted to Johannes Blömer for this observation). Indeed, every column of the matrix C is a linear combination of the vectors u_1, \dots, u_k , so $k \geq \text{rank } C$. On the other hand, any basis of the vector space generated by the columns of C can be chosen as u_1, \dots, u_k , because if we express every column of C as a linear combination of the u_i 's, then $v_{i,\nu}$ can be set to the coefficient at u_i for the column with index ν . \square

The above theorem implies that a linearization of the smallest possible dimension can be obtained efficiently by first computing the rank of the matrix C , say using the Gaussian elimination, and then expressing its columns in some base of the column space.

3. Range Spaces with Elementary Cells

In this section we begin discussing various methods recently developed for range searching and other problems, translating them into an abstract framework. See [14], [17], [19], [22], and [27] for some of the recent results on range searching.

Elementary Cells, Decompositions, Cuttings. Let (X, Γ) be a range space, and let \mathcal{E} be a set of subsets of X , which are called the *elementary cells*. The triple $S = (X, \Gamma, \mathcal{E})$ is called a *range space with elementary cells*, or shortly a *space*.

We say that an elementary cell $e \in \mathcal{E}$ *crosses* a range $\gamma \in \Gamma$ if $\gamma \cap e \neq \emptyset$ but $e \not\subseteq \gamma$. For $Q \subseteq \Gamma$ and $e \in \mathcal{E}$, we let Q_e denote the set of ranges of Q crossing e .

For a space $S = (X, \Gamma, \mathcal{E})$, we define the *derived range space* as

$$(\Gamma, \{\Gamma_e \mid e \in \mathcal{E}\}). \quad (3.1)$$

In our subsequent constructions we usually assume that the VC-dimension of the derived range space is bounded by a constant. In fact, this might be a good abstract definition capturing the essential properties expected from “constant complexity” cells. The assumption is satisfied in many natural cases, in particular, if both the ranges and the elementary cells are Tarski cells [33].

A *cutting* is any collection $\Xi \subseteq \mathcal{E}$ of disjoint elementary cells whose union is X . Let Q be a finite subset of Γ . A cutting Ξ is called a $(1/r)$ -cutting for Q if $|Q_e| \leq |Q|/r$ for every $e \in \Xi$. If $Q_e = \emptyset$ for every $e \in \Xi$, then Ξ is called an *elementary cell decomposition* for Q (with respect to S). We say that a range space with elementary cells (X, Γ, \mathcal{E}) has elementary cell decompositions of size $\xi(m)$ (ξ is a nondecreasing real function on natural numbers) if, for every $Q \subseteq \Gamma$ with $|Q| \leq m$, an elementary cell decomposition of Q consisting of at most $\xi(m)$ elementary cells exists.

These definitions generalize the corresponding notions for the range space defined by half-spaces, \mathcal{H}_d . Here $X = \mathbb{R}^d$, Γ is the set of all half-spaces in \mathbb{R}^d , and \mathcal{E} is the set of all relatively open simplices of all dimensions³ k , $0 \leq k \leq d$. With some abuse of notation, we use \mathcal{H}_d to denote this space as well. A simplex $e \in \mathcal{E}$ crosses a half-space γ if it has points both inside γ and outside γ . For a collection Q of half-spaces, any triangulation of the arrangement of the bounding hyperplanes is an elementary cell decomposition for Q . Thus, by well-known results on hyperplane arrangement triangulation, for any m -element Q , an elementary cell decomposition consisting of $O(m^d)$ cells exists [14].

For our range searching algorithm, we need $(1/r)$ -cuttings with a small number of elementary cells. The construction of $(1/r)$ -cuttings is well studied for the spaces \mathcal{H}_d , where the main result is as follows: For any finite set of half-spaces and any choice of the parameter r , a $(1/r)$ -cutting of size $O(r^d)$ exists, which is asymptotically optimal. This result was first proved by Chazelle and Fridman [14]. An efficient construction of $(1/r)$ -cuttings in this case has been studied (among others) in [11], [25], and [26].

³ We also admit “unbounded simplices” defined as intersections of at most $d + 1$ half-spaces.

Although computing an asymptotically optimal $(1/r)$ -cutting for the spaces \mathcal{H}_d is somewhat complicated [14], the existence of $(1/r)$ -cuttings can be established with only a slightly worse-size bound using a straightforward $(1/r)$ -nets argument (which probably was discovered independently by several authors, see [19] and [23]). Let \mathcal{H}' denote the derived range space for \mathcal{H}_d . It is easy to check that it has a bounded VC-dimension. Considering a finite set Q of half-spaces, clearly the subspace induced by Q in \mathcal{H}' also has a bounded VC-dimension, and thus by the above-quoted result on $(1/r)$ -nets it follows that we can choose a $(1/r)$ -net $N \subseteq Q$ for this range space, of size $O(r \log r)$. Let Ξ be any elementary cell decomposition for N with respect to \mathcal{H}_d (a triangulation in this particular case). The definition of $(1/r)$ -net and of the derived range space \mathcal{H}' imply that, for any simplex e not crossing any hyperplane of N , $|Q_e| \leq |Q|/r$, and therefore each simplex of Ξ crosses at most n/r hyperplanes of Q . Since Ξ covers \mathbb{R}^d , it is a $(1/r)$ -cutting for Q . We can choose Ξ of size $O(|N|^d) = O((r \log r)^d)$.

This argument is valid for any space (X, Γ, \mathcal{E}) , as long as the derived range space (3.1) has VC-dimension bounded by some constant D . Hence, we obtain

Lemma 3.1. *Let $S = (X, \Gamma, \mathcal{E})$ be a space such that the VC-dimension of the derived range space is bounded by some constant D . If S has elementary cell decompositions of size $\xi(m)$, then, for any r and Q , it admits $(1/r)$ -cuttings of size $\xi(C_D r \log r)$, where C_D is a constant.*

As mentioned above, the size of $(1/r)$ -cuttings for the spaces \mathcal{H}_d can be improved by more sophisticated methods (those of [11] or [14]). These methods can also be applied to some of the spaces that we use here. However, the savings by these methods is relatively small, so we do not consider such improvements in this paper.

Computational Assumptions. In what follows we assume that the space we work with admits a $(1/r)$ -cutting of at most $\zeta(r)$ elementary cells for every finite $Q \subseteq \Gamma$, where $\zeta(r)$ is a nondecreasing real function. In order to use this in algorithms we also have to make some computational assumptions. Namely, we assume that given r and (a suitable representations of) Q , a $(1/r)$ -cutting of size at most $\zeta(r)$ for Q can be computed in time $O(n \cdot F(r))$, where $n = |Q|$ and $F(r)$ is some function of r , thus, in particular, the cutting can be computed in linear time for $r = O(1)$.

This assumption is based on the above $(1/r)$ -net argument and a result of [25]. This results says that for suitably represented range spaces of n points, which bounded VC-dimension, a $(1/r)$ -net of size $O(r \log r)$ can be computed in $O(F_1(r)n)$ time. In our case we apply it to the derived range space (3.1) and compute the $(1/r)$ -net N as in the above argument. Having such an N , we have to compute an elementary cell decomposition for N with respect to the underlying range space. It seems reasonable to assume that this can be done in time depending on r only. It remains to say what is meant by “suitably represented” range space, but here we refer to [25] for a precise formulation of the assumption. We only note that if the range space in question admits a linearization of a bounded dimension (which is the case if both the ranges of Q and the elementary cells of \mathcal{E} are Tarski cells),

then the assumption is satisfied and so $(1/r)$ -cuttings for such spaces can be computed, provided that we can compute elementary cell decompositions.

Weighted Cuttings. We also need a “weighted” notion of cuttings. If Q is a set of ranges and $w: Q \rightarrow \mathbb{R}^+$ is a nonnegative real function on Q , we call the pair (Q, w) a *weighted collection of ranges*. If $A \subseteq Q$, we write just $w(A)$ for $\sum_{\gamma \in A} w(\gamma)$. A cutting Ξ is a *weighted $(1/r)$ -cutting for (Q, w)* if $w(Q_e) \leq w(Q)/r$ for every elementary cell $e \in \Xi$. A similar straightforward generalization is possible for the notion of a $(1/r)$ -net. Both the above argument for deducing the existence of small $(1/r)$ -cuttings using $(1/r)$ -nets and the algorithm for $(1/r)$ -net computation in [25] generalize to the weighted case. Hence, we may assume that $(1/r)$ -cuttings of size at most $\zeta(r)$ can be computed in $O(F(r)n)$ time even in the weighted sense.

4. Decomposing Arrangements of Algebraic Varieties

The efficiency of a range searching algorithm described below depends crucially on the size of the elementary cell decompositions of the underlying space. We are free to choose any suitable kind of elementary cells which help us to achieve this goal, provided that the derived range space defined in (3.1) has a bounded VC-dimension.

For ranges defined by polynomials of bounded degree, an elementary cell decomposition can be computed directly using the known results on decomposing arrangements of algebraic surfaces into Tarski cells. For $d = 2$, the arrangement of algebraic curves of bounded degree can be easily decomposed into Tarski cells by drawing a vertical line in both directions from each intersection point and the point of vertical tangency until it intersects another edge of the arrangement. See Fig. 1. For $d \geq 3$, the cylindrical algebraic decomposition scheme due to Collins can be used, but it produces too many cells. Recently, Chazelle *et al.* [12] proved that, given d -variate polynomials f_1, \dots, f_n of bounded degree, \mathbb{R}^d can be partitioned into $O(n^{2d-3}\beta(n))$ Tarski cells (where $\beta(n)$ is a function growing extremely slowly, more slowly than the inverse of any primitively recursive function), such

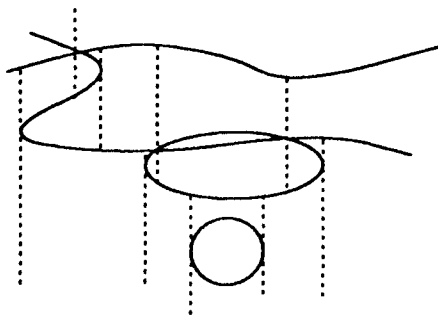


Fig. 1. Decomposing an arrangement of algebraic curves in the plane into Tarski cells.

that the sign of each f_i remains constant within each cell. For $d = 3$, this bound is almost optimal. For $d > 3$, this upper bound is significantly worse than the trivial (but best-known) lower bound of $\Omega(n^d)$.

Sometimes (for simple polynomials) better bounds are obtained using another method, which we outline next. In general, suppose we have some embedding φ of the considered range space (X, Γ) into a range space (Y, Σ) that has a natural elementary cell structure \mathcal{F} and a well-behaved elementary cell decomposition. We can define an elementary cell structure on (X, Γ) , in which the elementary cells are the preimages of the cells in \mathcal{F} , that is, $\mathcal{E} = \{\varphi^{-1}(f) \mid f \in \mathcal{F}\}$. An obvious candidate for the “target” space (Y, Σ, \mathcal{F}) is the space \mathcal{H}_k with a possibly small k . So, if φ is a linearization (X, Γ) of dimension k , the elementary cells of \mathcal{E} will be the preimages of at most k -dimensional simplices. We then get the existence of elementary cell decompositions of size $O(n^k)$ for (X, Γ, \mathcal{E}) .

Typically $\varphi(X)$ will be a d -dimensional (real) algebraic variety defined by bounded-degree polynomials, so the preimage of “most of” the simplices will be empty. We should therefore expect a better bound on the size of the elementary decomposition of (X, Γ, \mathcal{E}) . Namely, we can apply a recent result of Aronov *et al.* [8], claiming that a d -dimensional algebraic variety intersects at most $O(m^{\lfloor (d+k)/2 \rfloor} \log m)$ simplices in the canonical triangulation of an arrangement of m hyperplanes in \mathbb{R}^k . This means that when $\varphi(X)$ is a d -dimensional variety in \mathbb{R}^k , then the space $S = (X, \Gamma, \mathcal{E})$ has elementary cell decompositions of size $\xi(m) = O(m^{\lfloor (d+k)/2 \rfloor} \log m)$, which by Lemma 3.1 implies, that S admits $(1/r)$ -cuttings of size at most $\zeta(r) = O((r \log r)^{\lfloor (d+k)/2 \rfloor} \log r)$.

For example, the range space defined by balls on \mathbb{R}^d has a linearization of dimension $d + 1$, whose image is the unit paraboloid (see Section 2 for discussion of the case $d = 2$). By the above method elementary cell decompositions (or a decomposition of an arrangement of m spheres, in a more usual language) of size $O(m^d \log m)$ are obtained, which for $d \geq 4$ is the best-known result. Note that the resulting elementary cells in the original space are projections of intersections of simplices with the paraboloid, thus perhaps they are not the most natural cells to be considered for decomposing an arrangement of spheres.

5. Partition Theorem in an Abstract Setting

Throughout this section we let (X, Γ, \mathcal{E}) be a range space with elementary cells. For technical simplification we assume $X, \emptyset \in \mathcal{E}$. We begin by generalizing the notion of simplicial partition from [27].

Let $P \subseteq X$ be a set of n points. An *elementary cell partition* or shortly *partition* for P is a collection

$$\Pi = \{(P_1, e_1), \dots, (P_m, e_m)\},$$

where the P_i 's are disjoint subsets (called the *classes* of Π) forming a partition of P , and each $e_i \in \mathcal{E}$ (called an elementary cell of Π) contains P_i . The number m is the *size* of the partition.

For a range $\gamma \in \Gamma$, we define the *crossing number* of γ (relative to Π) as the number of elementary cells among e_1, \dots, e_m crossed by γ . The crossing number of Π is defined to be the maximum of crossing numbers over all $\gamma \in \Gamma$.

Theorem 5.1 (Partition Theorem). *Let $S = (X, \Gamma, \mathcal{S})$ be a space admitting $(1/t)$ -cuttings of size at most $\zeta(t)$. Suppose that (X, Γ) has a faithful linearization of dimension bounded by a constant. Let $P \subseteq X$ be a set of n points, and let r be a parameter, $1 < r < n$. Then an elementary cell partition $\Pi = \{(P_1, e_1), \dots, (P_m, e_m)\}$ for P exists, such that:*

- (i) $\lfloor n/r \rfloor \leq |P_i| < 2\lfloor n/r \rfloor$ for every i .
- (ii) The crossing number of Π is

$$O\left(\log r + \sum_{i=1}^r \frac{1}{\zeta^{-1}(i)}\right).$$

If r is bounded by a constant, and a $(1/t)$ -cutting of size at most $\zeta(t)$ can be computed in linear time for $t = O(1)$, then Π can be computed in $O(n)$ time.

Let us emphasize that in this theorem we assume a faithful linearization, that is, the preimage of any half-space is a range in Γ . We do not know whether this assumption is necessary, but it is required for the current proof. In general, a linearization of a range space is not faithful, so, in order to apply the theorem, we first have to add some “artificial” ranges to the range space (the preimages of half-spaces). Note that this “closure” is not a canonical one, since it depends on the chosen linearization. The elementary cell decompositions are needed for these artificial ranges as well.

A first ingredient of the proof is the following lemma:

Lemma 5.2. *Let $S = (X, \Gamma, \mathcal{E})$ be a space admitting $(1/t)$ -cuttings of size at most $\zeta(t)$. Let $P \subseteq X$ be a set of n points, let r be a parameter, $1 < r < n$, and let Q be a finite set of ranges of Γ . Then a partition for P , $\Pi = \{(P_1, e_1), \dots, (P_m, e_m)\}$, exists such that:*

- (i) $\lfloor n/r \rfloor \leq |P_i| < 2\lfloor n/r \rfloor$ for every i .
- (ii) The crossing number of every range of Q relative to Π is

$$O\left(\log |Q| + \sum_{i=1}^r \frac{1}{\zeta^{-1}(i)}\right).$$

Proof. The proof follows the one in [27]. Let $s = \lfloor n/r \rfloor$. We inductively construct the disjoint sets $P_1, P_2, \dots \subseteq P$, and elementary cells e_1, e_2, \dots ; $P_i \subseteq e_i$. Suppose that P_1, \dots, P_i have already been constructed, and suppose we want to compute P_{i+1} . Set $P'_i = P \setminus (P_1 \cup \dots \cup P_i)$, $n_i = |P'_i|$.

If $n_i < 2s$, we set $P_{i+1} = P'_i$, $e_{i+1} = X$, $m = i + 1$, and $\Pi = \{(P_1, e_1), \dots, (P_m, e_m)\}$, which completes the construction. If $n_i \geq 2s$, we proceed as follows. For a range $\gamma \in Q$, let $\kappa_i(\gamma)$ denote the number of elementary cells of $\{e_1, \dots, e_i\}$ crossed by γ . We define a weighted collection (Q, w_i) by setting $w_i(\gamma) = 2^{\kappa_i(\gamma)}$ for every $\gamma \in Q$.

Let us choose a parameter $t_i = \zeta^{-1}(n_i/s)$. Then a $(1/t_i)$ -cutting Ξ_i for (Q, w_i) of size at most n_i/s exists. By the pigeonhole principle, some of the elementary cells of the cutting Ξ_i contain at least s points of P'_i . Let e_{i+1} be some such elementary cell. P_{i+1} is an arbitrary subset of s points of P'_i contained in e_{i+1} . This completes the description of the construction.

Property (i) of Π is obvious from the construction. In order to bound the crossing numbers of the ranges of Q relative to the partition Π , we estimate the total weight of the ranges of Q after m steps, denoted $w_m(Q)$, in two different ways. By construction,

$$n_i = n - i \cdot s$$

and

$$t_i = \zeta^{-1}\left(\frac{n_i}{s}\right) = \zeta^{-1}\left(\frac{n}{s} - i\right) \geq \zeta^{-1}(r - i).$$

The weight $w_m(\gamma)$ of a range $\gamma \in Q$ that crosses κ elementary cells of Π is 2^κ . Therefore,

$$\kappa \leq \log_2 w_m(Q). \quad (5.1)$$

Let us consider how $w_{i+1}(Q)$ increases compared with $w_i(Q)$. The weight of ranges in $Q_{e_{i+1}}$ increases by a factor of two, and the weight of other ranges remains unchanged. Therefore,

$$w_{i+1}(Q) \leq w_i(Q) \left(1 + \frac{w_i(Q_{e_{i+1}})}{w_i(Q)}\right).$$

Since Ξ_i is a $(1/t_i)$ -cutting, we have

$$w_i(Q_{e_{i+1}}) \leq \frac{w_i(Q)}{t_i} \leq \frac{w_i(Q)}{\zeta^{-1}(r - i)}.$$

However, $w_0(Q) = |Q|$ and $m \leq \lfloor r \rfloor$, so we obtain

$$w_m(Q) \leq |Q| \prod_{i=0}^{r-1} \left(1 + \frac{1}{\zeta^{-1}(r - i)}\right) \leq |Q| \prod_{i=1}^r \left(1 + \frac{1}{\zeta^{-1}(i)}\right).$$

Now (5.1), along with the inequality $\ln(1 + x) \leq x$, yields

$$\kappa \leq \log_2 w_m(Q) = O\left(\log|Q| + \sum_{i=1}^r \frac{1}{\zeta^{-1}(i)}\right),$$

as required. \square

The second ingredient of the proof is the following lemma; it is the place where we need the linearization assumption.

Lemma 5.3 (Test Set Lemma). *Let (X, Γ, \mathcal{E}) be a space with a faithful linearization φ of dimension d bounded by a constant. Let $P \subseteq X$ be a set of n points, and let r ($1 < r < n$) be a parameter. Then a set $Q \subseteq \Gamma$ of size $O(r^d)$ exists such that, for any partition $\Pi = \{(P_1, e_1), \dots, (P_m, e_m)\}$ satisfying $|P_i| \geq n/r$ ($1 \leq i \leq m$), the following holds: If κ_0 is the maximum crossing numbers of ranges Q relative to Π , then the crossing number of Π is at most $(d + 1)\kappa_0$.*

Proof. Let $\varphi: X \rightarrow \mathbb{R}^d$ be the faithful linearization of (X, Γ) , set $P^* = \varphi(P)$. Let h be a hyperplane in \mathbb{R}^d , and let G be a finite set of hyperplanes in \mathbb{R}^d . We say that a point $p \in P^*$ lies in the *zone* of h (with respect to G) if p can be connected to a point of h by a segment that does not intersect any hyperplane of G . We call G an *s-guarding set* for h (relative to P^*) if there are less than s points of P^* in the zone of h . It is shown in [27] that, for any r and P^* , a set Q^* of $O(r^d)$ hyperplanes in \mathbb{R}^d exists such that there is an (n/r) -guarding set $G \subseteq Q^*$ of size $d + 1$ for every hyperplane h in \mathbb{R}^d .

Let us fix such a Q^* and, for every $h \in Q^*$, let \bar{h} denote one of the half-spaces bounded by h (arbitrarily chosen). We set $Q = \{\varphi^{-1}(\bar{h}) | h \in Q^*\}$, we claim that Q is the desired set of ranges.

Suppose that every range of Q has crossing number at most κ_0 with respect to a partition Π as in the lemma, and let $\gamma \in \Gamma$ be a range. Let h be the hyperplane bounding the half-space $\varphi^*(\gamma)$, and let $G = \{g_1, \dots, g_{d+1}\}$ be the (n/r) -guarding set for h . Consider an elementary cell e_i of the partition Π that crosses the range γ . Let P_i be the class corresponding to e_i . It is enough to show that $\varphi(e_i)$ crosses some half-space \bar{g} for a $g \in G$, because then e_i crosses the preimage $\varphi^{-1}(\bar{g}) \in Q$, and the preimages of the at most $d + 1$ half-spaces of G are crossed by at most $(d + 1)\kappa_0$ elementary cells in total.

Suppose the contrary; then $\varphi(e_i)$ is contained in a single cell of the arrangement of G . Since $\varphi(e_i)$ contains at least n/r points of P^* (the images of the points of P_i), this cell cannot be incident to h , otherwise there would be n/r points of P^* in the zone of h , contradicting the definition of the (n/r) -guarding set. This means that the whole $\varphi(e_i)$ lies in one of the half-spaces defined by h , but this contradicts the fact that e_i crosses $\varphi^{-1}(\bar{h})$. \square

Proof of Theorem 5.1. Combining Lemma 5.3 with Lemma 5.2, we obtain the existence result in the Partition Theorem, Theorem 5.1. As for the algorithmic

statement, we note that if r is bounded by a constant, the number of steps in the construction in Lemma 5.2 is bounded by a constant, and each of the steps can be executed in at most $O(n)$ time. The set Q^* in the proof of the Test Set Lemma (Lemma 5.3) can also be obtained in $O(n)$ time. This finishes the proof of Theorem 5.1. \square

6. Range Searching and Related Results

Γ -Range Searching Problem. We now construct data structures for the Γ -range searching problem based on Theorem 5.1.

Theorem 6.1. *Let $f(x_1, \dots, x_d, a_1, \dots, a_p)$ be a $(d + p)$ -variate polynomial. Assume that $p, d, q, \deg(f)$ are bounded by a constant. Let*

$$\Gamma = \{ \{x \in \mathbb{R}^d \mid f(x, a^1) \geq 0, \dots, f(x, a^q) \geq 0\} \mid a^1, \dots, a^q \in \mathbb{R}^p \}.$$

Then the Γ -range searching problem with semigroup weights can be solved with $O(n)$ space, $O(n \log n)$ preprocessing time, and $O(n^{1-1/b+\delta})$ query time, where the parameter b can be bounded as follows:

- (i) $b = d$ for $d \leq 3$, $b \leq 2d - 3$ for $d > 3$.
- (ii) *If f can be written in the form*

$$f(x, a) = \psi_0(a) + \psi_1(a)\varphi_1(x) + \dots + \psi_k(a)\varphi_k(x), \quad (6.1)$$

where $\psi_0, \dots, \psi_k, \varphi_1, \dots, \varphi_k$ are bounded-degree polynomials, then

$$b \leq \left\lfloor \frac{d+k}{2} \right\rfloor.$$

In particular, for $d = 2, 3$, we get $O(n^{1/2+\delta})$ (resp. $O(n^{2/3+\delta})$) query time, which is close to optimal.

Proof. Here we only prove part (i). Part (ii) is contained in Theorem 6.3 below, as Γ -range searching, in this case, is equivalent to range searching with intersections of q half-spaces on the set $\varphi(P)$.

We first verify the applicability of Theorem 5.1 in the described situation. Then we build the data structure by a standard partition tree construction.

Let φ be a linearization of a bounded dimension k of the range space (\mathbb{R}^d, Γ_f) (the ranges in Γ_f are defined by a *single* polynomial inequality, see (1.1)). We use the methods outlined in Section 2 to obtain such a linearization. In order to apply the Partition Theorem (Theorem 5.1), we need to make φ faithful. We let $\bar{\Gamma}$ be the set of all φ -preimages of half-spaces in \mathbb{R}^k . Then φ is a faithful linearization of the range space $(\mathbb{R}^d, \bar{\Gamma})$.

The next step is to define elementary cells for $(\mathbb{R}^d, \bar{\Gamma})$. Observe that the ranges

of $\bar{\Gamma}$, albeit more general than the ones of Γ_f , are still defined by a single polynomial inequality of bounded degree. Hence we can invoke the result of [12], that is, an arrangement of n algebraic surfaces in \mathbb{R}^d , each of a constant maximum degree, can be decomposed into $O(n^b \beta(n))$ Tarski cells, with $b = 2d - 3$ (for $d \geq 3$). Let \mathcal{E} be the set of all the cells that can arise in such decompositions for arrangements of surfaces bounding the ranges of $\bar{\Gamma}$. Then the space $(\mathbb{R}^d, \bar{\Gamma}, \mathcal{E})$ has elementary cell decompositions of size $\xi(m) = O(m^b \beta(m))$, and thus $(1/r)$ -cuttings of size $\zeta(r) = O(r^b \log^b r \beta(r)) = O(r^b \log^{b+1} r)$.

In view of the above discussion, we can apply the Partition Theorem (Theorem 5.1). Thus, for a given set P of n points in \mathbb{R}^d and a parameter $r < n$, an elementary cell partition Π exists with class size between $\lfloor n/r \rfloor$ and $2\lfloor n/r \rfloor$, and with crossing number

$$O\left(\log r + \sum_{i=1}^r \frac{1}{i^{1/b} \log^{1+1/b} i}\right) = O(r^{1-1/b} (\log r)^{1+1/b}).$$

If $r = O(1)$, Π can be computed in $O(n)$ time.

These partitions are guaranteed to have small crossing numbers with respect to ranges of Γ_f (actually for ranges of $\bar{\Gamma}$, but these only play an auxiliary role in the construction and will not be significant anymore). We observe that this also gives small crossing numbers for ranges of Γ : Let $\gamma = \gamma_1 \cap \dots \cap \gamma_s \in \Gamma$, $\gamma_j \in \Gamma_f$ and $s = O(1)$, then it is easily seen that an elementary cell e crossing γ must cross at least one γ_j , and hence the crossing number of γ is at most s times the maximum crossing number of the ranges of Γ_f .

We are now ready to construct the data structure—partition tree—for Γ -range searching on P . Each node v of the tree will correspond to some subset $P_v \subseteq P$, the root will correspond to the whole P . Let $n_v = |P_v|$. The sets associated with the children of a node v will form a partition of P_v . For each leaf v , n_v is bounded by some suitable constant.

We construct the partition tree in a top-down fashion starting from the root. We let r be a sufficiently large constant, which remains fixed throughout the construction. We compute an elementary cell partition Π_v for P_v with class sizes between $\lfloor n_v/r \rfloor$ and $2\lfloor n_v/r \rfloor$, such that any range of Γ crosses at most $\kappa = Cr^{1-1/b} \log^{1+1/b} r$ elementary cells of Π_v , C is a constant independent of r . The classes in Π_v correspond to the (at most r) children of v . With each node, we store the description of the elementary cells of Π_v and the total weight of the points in each class of Π_v (but not the list of points of P_v ; the points will only be stored in the leaves). This partition tree occupies linear space and can be constructed in $O(n \log n)$ time.

When processing a query with a range $\gamma \in \Gamma$, we start at the root. At each node v visited, we take the elementary cells of the Π_v one by one. The cells contained in γ or disjoint from γ are handled directly, and we recursively search at the children of v corresponding to elementary cells that cross γ . We get the following recurrence for the query time $T(n)$:

$$T(n) \leq O(r) + \kappa T\left(2\left\lfloor \frac{n}{r} \right\rfloor\right),$$

with initial condition $T(n) = O(1)$ for n smaller than some constant. The solution is $T(n) = O(n^{1-1/(b+\delta)})$, with $\delta \rightarrow 0$ as $r \rightarrow \infty$. \square

Remark 6.2. The above theorem is formulated for ranges defined as the conjunction of a constant number of d -variate polynomial inequalities. Each polynomial is a restriction of the *same* $(d+p)$ -variate polynomial $f(x_1, \dots, x_d, a_1, \dots, a_p)$ (evaluated at a_1, \dots, a_p); it might be asked what happens if ranges are defined by several *different* polynomials. That is, Γ is of the form

$$\Gamma = \{ \{x \in \mathbb{R}^d \mid f_1(x, a^1) \geq 0, \dots, f_q(x, a^q) \geq 0\} \mid a^1, \dots, a^q \in \mathbb{R}^p \},$$

where each $f_i(x, a)$ is a $(d+p)$ -variate polynomial. One (rather artificial) possible approach is to convert this case to the previous one by defining a polynomial g (with possibly more parameters) which is an extension of f_1, \dots, f_q . For example, if $f_1(x, a), f_2(x, a)$ are two $(d+p)$ -variate polynomials, we can write $f_1(x, a) = g(x, a, 0)$, $f_2(x, a) = g(x, a, 1)$ for $g(x, a, z) = (1-z)f_1(x, a) + zf_2(x, a)$, where z is a new parameter. If we use case (i) of the theorem, this extension does not even affect the asymptotic efficiency of the algorithm. Another solution, which is sometimes more suitable in case (ii), is to construct a multilevel data structure as mentioned in the introduction.

Simplex Searching for Points on an Algebraic Variety. Another instance of range searching amenable to our methods is the simplex range searching for point sets P lying on an algebraic variety V of bounded degree in \mathbb{R}^k . Here the general simplex range searching result can be improved. We have

Theorem 6.3. *Let P be a set of n points in \mathbb{R}^k lying on a d -dimensional algebraic variety defined by bounded-degree polynomials. Then the simplex range searching problem for P can be solved with linear space, $O(n \log n)$ preprocessing time, and $O(n^{1-1/(b+\delta)})$ query time, where $b = \lfloor (d+k)/2 \rfloor$.*

This result includes Theorem 6.1(ii): the set $V = \varphi(\mathbb{R}^d)$ in this case is a d -dimensional algebraic variety, and, since every range of Γ can be expressed as a preimage of an intersection of q half-spaces in \mathbb{R}^k (which, in turn, is a disjoint union of $O(1)$ simplices), simplex range searching on the set $\varphi(P) \subset V$ may be applied.

Proof. We define a space with elementary cells (V, Γ, \mathcal{E}) by letting Γ be the set of intersections of V with all half-spaces and let \mathcal{E} be the set of all intersections of V with simplices in \mathbb{R}^k . By the zone theorem of Aronov *et al.* [8], this space admits elementary cell decompositions of size $O(m^b \log m)$, and hence $(1/r)$ -cuttings of size $O(r^b (\log r)^{b+1})$ (both the ranges and elementary cells are Tarski cells in \mathbb{R}^k). The inclusion map $V \rightarrow \mathbb{R}^k$ is, by definition, a faithful linearization. Hence we can apply Theorem 5.1, and we can build a partition tree for a given set $P \subset V$ as in the previous theorem. \square

7. Other Applications

In the last section we constructed an efficient data structure for Γ -range searching using Theorem 5.1. It has been shown in [1], [5], [9], and in other papers that the simplex range searching data structures are quite general, and can be applied to various other problems involving a collection of linear objects. Similarly, the partition tree developed in the previous section can yield fast data structures for a number of problems involving semialgebraic sets. It is beyond the scope of this paper to discuss all the applications, so we have chosen two sample applications.

7.1. Ray Shooting Among Triangles in Three Dimensions

The first application that we consider is the three-dimensional *ray-shooting* problem, which is defined as follows: Preprocess a set Δ of n triangles in \mathbb{R}^3 into a data structure, so that the first triangle of Δ hit by a query ray can be reported quickly. In the last few years ray shooting has received considerable attention due to its applications in computer graphics and other geometric problems, e.g., see [1], [4], [5], [9], [15], [23], and [30]. The best-known algorithm for ray shooting in three dimensions is due to Agarwal and Sharir [5] that can answer a query in $O(n^{4/5})$ time using $O(n^{1+\epsilon})$ space and preprocessing. If $O(n^{4+\delta})$ space and preprocessing are allowed, a query can be answered in $O(\log n)$ time [5], [9], [30]. In this subsection we show that the query time can be improved to close to $O(n^{3/4})$ using linear space and $O(n \log n)$ time preprocessing.

A result by Agarwal and Matoušek [4] shows that it suffices to construct a data structure for detecting an intersection between Δ and a query segment. (Their result requires that the segment intersection detection algorithm satisfies some mild assumptions. These assumptions are satisfied in our case; see [4] for details.) Observe that a segment γ intersects a triangle \triangle if and only if the following two conditions are satisfied:

- (i) The line containing γ intersects \triangle .
- (ii) γ intersects the plane containing \triangle .

We construct a data structure on Δ that detects the above two conditions for a query line. The overall approach is similar to that of Agarwal and Sharir, so we only describe the main idea.

Let $\triangle \in \Delta$ be a triangle with edges e_1, e_2, e_3 . Let l_1, l_2 , and l_3 be the lines containing these edges. We orient the lines so that \triangle lies to the right of each of them. A line λ intersects \triangle if and only if it has the same relative orientation with respect to l_1, l_2 , and l_3 . The relative orientation of two oriented lines l, λ in \mathbb{R}^3 is defined to be the orientation of any simplex $abcd$, where $a, b \in l, c, d \in \lambda$, so that l is oriented from a to b and λ is oriented from c to d . Equivalently, it is also the sign of the inner product between the two vectors in projective \mathbb{R}^5 representing the *Plücker coordinates* of the two lines. (For the sake of convenience, we do not distinguish between the projective 5-space and the affine 5-space \mathbb{R}^5 .) To be more

precise, l can be mapped to a point $\pi(l)$, called a *Plücker point*, and λ can be mapped to a hyperplane $\varpi(\lambda)$, called a *Plücker hyperplane* in \mathbb{R}^5 , so that l has positive orientation with respect to λ if and only if $\pi(l)$ lies in the positive half-space bounded by the hyperplane $\varpi(\lambda)$. The Plücker points of all lines in \mathbb{R}^3 lie on a quadric surface, known as the *Plücker surface*, in \mathbb{R}^5 . See [13] and [32] for more details on Plücker's coordinates.

Based on the above approach we preprocess Δ as follows. We take an edge from each triangle of Δ and form the collection of lines containing these edges. The lines are oriented as described above, and are mapped to a collection P of points in \mathbb{R}^5 using Plücker coordinates. Let r be some sufficiently large constant. We construct a partition $\Pi = \{(P_1, e_1), (P_2, e_2), \dots, (P_m, e_m)\}$ for P of size $O(r)$ using Theorem 5.1. Since P lies on a four-dimensional quadratic surface, the crossing number of Π is $\kappa = O(r^{3/4} \log^{5/4} r)$. We recursively construct a partition for each canonical subset P_i of P . The recursion stops when the number of points fall below some prespecified constant. The resulting data structure is a tree of depth $O(\log n)$, each of whose node has degree at most r . Every node v of the tree is associated with a simplex e_v and a subset $P_v \subseteq P$. We take the second edge of the triangles corresponding to points in P_v , extend them to oriented lines, map the lines to their Plücker points in 5-space, and construct a partition tree on the resulting points as above. We attach this partition tree to v as its secondary structure. Next, for each canonical subset of the secondary tree, we take the third edge of the corresponding triangles and again repeat the same step. This gives a three-level structure, which is used to extract the triangles that intersect the line containing a query segment.

Let T be the set of triangles corresponding to a canonical subset of the third-level tree. We extend the triangles of T to full planes and dualize them to points. Let A be the set of resulting points. We preprocess A , in time $O(|A| \log |A|)$, into a linear-size data structure for answering simplex range queries [27]. This completes the description of the data structure. Let $S^i(m)$ denote the maximum space required by the i th level ($i \leq 4$) data structure constructed on m points. Then we get the following recurrence:

$$S^i(m) \leq \begin{cases} O(1) & \text{if } m \leq c, \\ O(m) & \text{if } m > c, \quad i = 4, \\ \sum_j S^i(m_j) + S^{i+1}(m) & \text{if } m > c, \quad i < 4, \end{cases}$$

where $\sum_j m_j = m$, $m_j \leq 2m/r$, and c is some appropriate constant. The solution of the above recurrence is $O(m \log^{4-i} m)$. The total space required by the above data structure is thus $O(n \log^3 n)$. Similarly, the total preprocessing time can be shown to be $O(n \log^4 n)$.

Let γ be a query segment. We detect an intersection between γ and Δ as follows. We first determine the triangles that intersect the line l containing γ and then check whether γ intersects any of the planes containing these triangles. Let h be the Plücker hyperplane of l (l is oriented arbitrarily). We search the first-level

structure with h and at each node v we do the following. If v is leaf, then we explicitly check (in constant time) whether γ intersects any of the triangles corresponding to the points associated with v . If v is not a leaf and the simplex e_v intersects h , we recursively search the subtree rooted at v . If h lies above (resp. below) e_v , we query the secondary structure of v with the half-plane lying below (resp. above) h .

Suppose e_v lies in the half-space h^+ . Let w be a node in the secondary structure. If w is a leaf, then we check whether γ intersects any of the triangles corresponding to the points associated with w . If h^+ does not intersect e_w , we discard w and do not search the subtree rooted at w . If h intersects w , we recursively search the subtree rooted at w . Otherwise, $e_w \subseteq h^+$, and we search the third-level tree stored at w in the same way as the second-level structure. Suppose z is a node of the third-level tree for which $e_z \subseteq h^+$, so that we query the fourth-level structure. Let T be the set of triangles corresponding to the points associated with z . By the above discussion, l intersects all triangles in T , so it suffices to determine whether γ intersects any of the planes containing the triangles of T . To this end, we query the fourth-level structure stored at z with the double wedge γ^* , dual to the segment γ , and determine whether any point (dual to planes containing the triangles of T) in the fourth-level structure lies in γ^* . If the answer is “yes,” then we can conclude that γ intersects a triangle of T , and stop right away. Otherwise, we continue. If the answer is “no” at all third-level nodes visited by the procedure, then γ does not intersect any triangle of Δ . This completes the description of the query-answering procedure.

The time spent at a fourth-level structure constructed on T is $O(|T|^{2/3+\delta})$. Let $Q^i(m)$ denote the maximum query time for an i th level structure constructed on m points. Then

$$Q^i(m) \leq \begin{cases} O(1) & \text{if } m \leq c, \\ O(m^{2/3+\delta}) & \text{if } m > c, \quad i = 4, \\ \kappa Q^i\left(\frac{2m}{r}\right) + Q^{i+1}(m) + O(1) & \text{if } m > c, \quad i < 4, \end{cases}$$

where c, c' are some constants. The solution of the above recurrence is $O(m^{3/4+\delta'})$ for another but arbitrarily small $\delta' > 0$. Hence, the overall query time is $O(n^{3/4+\delta'})$.

The size of the data structure can be improved to $O(n)$ without affecting the asymptotic query time by using the techniques described in Agarwal *et al.* [3]. Furthermore, using the standard space/query-time tradeoff techniques, the query time can be improved by allowing more space. In particular, combining our result with the result of Agarwal and Sharir [5], a segment intersection detection query can be answered in $O(n/s^{1/4})$ time using $O(s^{1+\delta})$ space and preprocessing for $n \leq s \leq n^4$. Putting everything together, we can conclude

Theorem 7.1. *Given a set Δ of triangles in \mathbb{R}^3 and a parameter s , $n \leq s \leq n^4$, Δ can be preprocessed into a data structure of size $O(s^{1+\delta})$, so that a ray-shooting query can be answered in time $O(n/s^{1/4})$.*

Remark 7.2. Pellegrini [30] has considered several other problems involving lines in three dimensions. Using the same approach as above, data structures for these problems can be obtained with the same performance as in Theorem 7.1.

7.2. Intersection Detection Between Lines and Spheres

Another application that we describe is: Preprocess a set \mathcal{S} of spheres in \mathbb{R}^3 , so that it can be quickly determined whether a query line intersects any of the spheres in \mathcal{S} .

Without loss of generality we can assume that the query line is not horizontal, because a similar (actually somewhat simpler) data structure can be constructed to handle horizontal lines. A nonhorizontal line a can be parametrized by a 4-tuple (a_1, a_2, a_3, a_4) of real numbers:

$$a = \{p + tq; t \in \mathbb{R}\},$$

where $p = (a_1, a_2, 0)$ is the intersection point of a with the xy -plane and $q = (a_3, a_4, 1)$ is the direction vector of a . The distance of the line a from a point $x = (x_1, x_2, x_3) \in \mathbb{R}^3$ is the same as the distance of the line $a' = \{(p - x) + tq; t \in \mathbb{R}\}$ from the origin. The point y on a' closest to the origin satisfies $y = (p - x) + tq$ for some t , and at the same time $y \cdot q = 0$. From this we get

$$y = (p - x) + \frac{[(p - x) \cdot q]q}{\|q\|^2},$$

and the desired distance of x from a will be the length of this vector y . After some calculation we obtain that the line a intersects a sphere with center (x_1, x_2, x_3) and radius r if and only if $f(x_1, x_2, x_3, r, a_1, a_2, a_3, a_4) \leq 0$, where

$$\begin{aligned} f(x_1, x_2, x_3, x_4, a_1, a_2, a_3, a_4) &= [(a_4^2 + 1)a_1^2 + (a_3^2 + 1)a_2^2 - 2a_1a_2a_3a_4] \\ &\quad + 2[a_2a_3a_4 - a_1(a_4^2 + 1)]x_1 + 2[a_1a_3a_4 - a_2(a_3^2 + 1)]x_2 \\ &\quad + 2[a_1a_3 + a_2a_4]x_3 - 2[a_3a_4]x_1x_2 - 2[a_3]x_1x_3 - 2[a_4]x_2x_3 \\ &\quad + [1](x_1^2 + x_2^2 - x_4^2) + [a_3^2](x_2^2 + x_3^2 - x_4^2) + [a_4^2](x_1^2 + x_3^2 - x_4^2). \end{aligned} \quad (7.1)$$

A sphere $S \in \mathcal{S}$ with parameters (x_1, x_2, x_3, x_4) can be mapped to a point $S^* = (x_1, x_2, x_3, x_4) \in \mathbb{R}^4$, and a nonhorizontal line a with parameters (a_1, a_2, a_3, a_4) can be mapped to a point $a^* = (a_1, a_2, a_3, a_4) \in \mathbb{R}^4$. Then a intersects S if and only if $S^* \in \gamma_f(a^*)$, thus intersection detection reduces to Γ_f -range searching (emptiness queries, in fact). Since $d = 4$, by Theorem 6.1, the latter query can be answered in time $O(n^{4/5+\delta})$.

However, here is a better way to proceed: We choose a linearization φ of the range space Γ_f . The polynomial in (7.1) is written in the form (2.1), which gives a linearization of dimension 9. A line a intersects a sphere S if and only if $\varphi(S^*)$ lies in the half-space $\varphi^*(a^*)$. Hence, the problem can be viewed as the half-space emptiness problem in dimension 9 (see Section 1). The latter can be solved with $O(n \log n)$ preprocessing, $O(n)$ space, and $O(n^{3/4} \log^{O(1)} n)$ query time, by [28]. We thus get:

Theorem 7.3. *We can preprocess a set \mathcal{S} of n spheres in \mathbb{R}^3 , in time $O(n \log n)$, into a data structure of linear size, so that it can be determined in time $O(n^{3/4} \log^{O(1)} n)$ whether a query line intersects any sphere of \mathcal{S} .*

Remark 7.4. (i) Using multilevel data structures and the technique of [4], the above algorithm can be extended to ray shooting among spheres with similar performance bounds; we omit the details.

(ii) Using the data structure described in [18], instead of the one described in [28], for answering half-space emptiness queries, a set of spheres in \mathbb{R}^3 can be preprocessed into a data structure of size $O(n^{4+\delta})$, so that a ray-shooting query can be answered in time $O(\log^2 n)$. Following a different approach, Agarwal *et al.* [2] have also obtained similar bounds. However, their approach does not yield a linear-size data structure with $O(n^{3/4} \log^{O(1)} n)$ query time. Combining Theorem 7.3 with these data structures, a ray shooting query can be answered in time $O(n/s^{1/4} \log^2 n)$ using $O(s^{1+\delta})$ space and preprocessing, for any $n \leq s \leq n^4$.

(iii) Consider an analogous problem in the plane—preprocess a set of circles into a data structure so that an intersection between a query line and circles can be detected. Here we get $O(n^{2/3+\delta})$ query time by a direct application of Theorem 6.1 (first we map circles to points in \mathbb{R}^3 using (2.4), then the problem becomes range searching with ranges defined by a single polynomial). Notice that a line $y = a_1x + a_2$ intersects a circle with center (x_1, x_2) and radius r if the polynomial $f(x_1, x_2, r, a_1, a_2) \geq 0$, where

$$\begin{aligned} f(x_1, x_2, x_3, a_1, a_2) = & [a_2^2] + 2[a_1a_2]x_1 - 2[a_2]x_2 + [a_1^2](x_1^2 - x_3^2) \\ & - 2[a_1]x_1x_2 + (x_2^2 - x_3^2). \end{aligned} \quad (7.2)$$

This polynomial has a linearization of dimension 5, as shown above, so by a five-dimensional half-space range searching structure [28], a line-circle intersection detection query can be answered in time $O(\sqrt{n} \log^{O(1)} n)$. A similar bound was attained in [3] using an entirely different approach.

(iv) A more interesting case is where the roles of lines and circles are reversed: A set of lines in the plane is to be preprocessed, so that it can be quickly detected whether a query circle intersects any of the given lines. Here the two approaches— Γ -range searching in the plane and half-space range searching in higher dimensions—yield similar results. Since a line is represented by two parameters, the first approach (see Theorem 6.1) can answer a query in time $O(n^{1/2+\delta})$. On the other hand, the polynomial of (7.2) has a linearization of dimension 5 in this

case as well, so a five-dimensional half-space range searching structure yields roughly the same query time. However, the first solution has certain advantages: It can be extended to counting the number of lines intersected by a query circle (while half-space range searching only works for emptiness queries or reporting), and it is possible to use the first structure as the top level of a multilevel data structure, which does not seem possible with the half-space range searching data structure. Moreover, the asymptotic efficiency of the first structure does not change even if we allow more complex query objects, e.g., ellipse, parabola, or a lune (intersection of two disks), because its efficiency depends only on the number of parameters required to represent a line.

8. Open Problems

One of major problems in computational geometry is the decomposition of an arrangement of algebraic varieties into Tarski cells. The known results for this problem, discussed in Section 4, appear to be far from being optimal. Our paper adds one more motivation (efficient range searching) to this problem, but various other applications are also known (e.g., point location; see [12]).

Concerning our results on range searching with linear space, if the query time should be significantly smaller than for the trivial solution (testing every point of P for membership in the query range) for some reasonable values of n , the constant b has to be quite small (certainly smaller than 10). Therefore it makes sense to investigate specific small dimensions and improve the decompositions of arrangements of algebraic surfaces at least in various special cases.

For many interesting range searching problems, our technique requires adding some new ranges (the preimages of half-spaces in the chosen linearization) before decomposition takes place, which makes the decomposition problem potentially more demanding. Is there a way to avoid these “artificial” ranges?

An interesting question is whether the half-space range searching algorithm of [28] can be generalized to nonlinear ranges in a similar way, as we did with the simplex range searching in this paper. In this case the underlying decomposition problem requires decomposing a single cell in an arrangement of algebraic varieties. We are not aware of any general results in higher dimensions in this direction, where the complexity of the single-cell decomposition is better than the complexity of the decomposition of the whole arrangement.

Another, slightly less-related problem is to prove (or disprove?) the obvious-looking statement that the range space defined (say) by triangles in the plane has no linearization. In general, perhaps more should be learnt about the embeddability of geometrically defined range spaces.

Acknowledgments

The authors are grateful to Johannes Blömer for the observation in the proof of Theorem 2.1, and to Micha Sharir for several useful comments.

References

1. P. K. Agarwal. Ray shooting and other applications of spanning trees with low stabbing number. *SIAM J. Comput.* **21** (1992), 540–570.
2. P. K. Agarwal, L. Guibas, M. Pellegrini, and M. Sharir. Ray shooting among spheres. Manuscript, 1993.
3. P. K. Agarwal, M. van Kreveld, and M. Overmars. Intersection queries for curved objects. *J. Algorithms* **15** (1993), 229–266.
4. P. K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.* **22** (1993), 794–806.
5. P. K. Agarwal and M. Sharir. Applications of a new space partitioning scheme. *Discrete Comput. Geom.* **9** (1993), 11–38.
6. A. Aggarwal, M. Hansen, and T. Leighton. Solving query-retrieval problems by compacting Voronoi diagrams. *Proc. 21st ACM Symposium on Theory of Computing*, 1990, pp. 331–340. Also to appear in *J. Assoc. Comput. Mach.*
7. N. Alon, D. Haussler, E. Welzl, and G. Wöginger. Partitioning and geometric embedding of range spaces of finite Vapnik–Chervonenkis dimension. *Proc. 3rd ACM Symposium on Computational Geometry*, 1987, pp. 331–340.
8. B. Aronov, M. Pellegrini, and M. Sharir. On the zone of a surface in a hyperplane arrangement. *Discrete Comput. Geom.* **9** (1993), 177–186.
9. M. de Berg, D. Halperin, M. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray shooting and hidden surface removal. *Proc. 7th Symposium on Computational Geometry*, 1991, pp. 51–60.
10. B. Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.* **2** (1989), 637–666.
11. B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.* **9** (1993), 145–158.
12. B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. A singly-exponential stratification scheme for real semi-algebraic varieties and its applications. *Theoret. Comput. Sci.* **84** (1991), 77–105. (Also in *Proc. 16th International Colloquium on Automata, Languages, and Programming*, 1989, pp. 179–192.)
13. B. Chazelle, H. Edelsbrunner, L. Guibas, M. Sharir, and J. Stolfi. Lines in space: combinatorics and algorithms. *Proc. 21st ACM Symposium on Theory of Computing*, 1989, pp. 382–393. Also to appear in *Algorithmica*.
14. B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica* **10** (1990), 229–249.
15. B. Chazelle and L. Guibas. Visibility and intersection problems in plane geometry. *Discrete Comput. Geom.* **4** (1989), 551–589.
16. B. Chazelle, M. Sharir, and E. Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica* **8** (1992), 407–430.
17. B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.* **4** (1989), 467–490.
18. K. L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.* **17** (1988), 830–847.
19. K. L. Clarkson and P. Shor. New applications of random sampling in computational geometry, II. *Discrete Comput. Geom.* **4** (1989), 387–421.
20. D. Dobkin and H. Edelsbrunner. Space searching for intersecting objects. *J. Algorithms* **8** (1987), 348–361.
21. H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Berlin, 1987.
22. D. Haussler and E. Welzl. ϵ -nets and simplex range queries. *Discrete Comput. Geom.* **2** (1987), 127–151.
23. J. Hershberger and S. Suri. A pedestrian approach to ray shooting: shoot a ray, take a walk. *Proc. 4th ACM–SIAM Symposium on Discrete Algorithms*, 1993, pp. 54–63.
24. J. Komlós, J. Pach, and G. Wöginger. Almost tight bounds for epsilon-nets. *Discrete Comput. Geom.* **7** (1992), 163–173.

25. J. Matoušek. Approximations and optimal geometric divide-and-conquer. *Proc. 23rd ACM Symposium on Theory of Computing*, 1991, pp. 506–511.
26. J. Matoušek. Cutting hyperplane arrangements. *Discrete Comput. Geom.* **6** (1991), 385–406.
27. J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.* **8** (1992), 315–334.
28. J. Matoušek. Reporting points in half-spaces. *Comput. Geom. Theory Appl.* **2** (1992), 169–186.
29. J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.* **10** (1993), 183–196.
30. M. Pellegrini. Ray shooting in 3-dimensional spaces. *Algorithmica* **9** (1993), 471–494.
31. F. Preparata and M. I. Shamos. *Computational Geometry—An Introduction*. Springer-Verlag, New York, 1985.
32. D. Sommerville. *Analytical Geometry in Three Dimensions*. Cambridge University Press, Cambridge, 1951.
33. R. Wenocur and R. Dudley. Some Vapnik–Chervonenkis classes. *Discrete Math.* **33** (1981), 313–318.
34. D. E. Willard. Polygon retrieval. *SIAM J. Comput.* **11** (1982), 149–165.
35. F. F. Yao and A. C. Yao. A general approach to geometric queries. *Proc. 17th ACM Symposium on Theory of Computing*, 1985, pp. 163–168.

Received October 13, 1992, and in revised form June 11, 1993.