Algorithmica (2004) 38: 145–160
DOI: 10.1007/s00453-003-1047-0

Algorithmica

Covering with Ellipses[1]

Alon Efrat,[2] Frank Hoffmann,[3] Christian Knauer,[3]
Klaus Kriegel,[3] Günter Rote,[3] and Carola Wenk[2]

Abstract. We address the problem of how to cover a set of *required points* by a small number of *axis-parallel ellipses* that avoid a second set of *forbidden points*. We study geometric properties of such covers and present an efficient randomized approximation algorithm for the cover construction. This question is motivated by a special pattern recognition task where one has to identify ellipse-shaped protein spots in two-dimensional electrophoresis images.

Key Words. Algorithms and data structures, Computational geometry, Approximation algorithm, Set cover, Proteomics.

1. Introduction and the Application Background.

In this paper we develop an efficient randomized approximation algorithm for the following problem:

THE GENERAL ELLIPSE COVERING PROBLEM. Given a set F of *n forbidden* points and a set R of *m required* points, find a set $\mathbf{E} = \{E_1, \ldots, E_k\}$ of axis-parallel ellipses, of minimal cardinality k, such that their union $\bigcup \mathbf{E} := \bigcup_{E \in \mathbf{E}} E$ *covers* R and *strictly respects* F, i.e., $R \subseteq \operatorname{int}(\bigcup \mathbf{E})$ and $F \cap \operatorname{int}(\bigcup \mathbf{E}) = \emptyset$. Thus $\cup\mathbf{E}$ has to contain fully R in its interior and may contain no points from F except on its boundary.

The lower right part of Figure 1 shows a set of 43 required points (black) and 24 forbidden points (white) forming a subset of the grid, and a cover by four ellipses.

Motivation. This problem stems from a pattern recognition task in proteomics, which is a rapidly growing field within molecular biology. In proteomics two-dimensional electrophoresis (2DE) is a well known and widely used technique to separate the protein components of a probe. A 2DE gel is the product of two separations performed sequentially in acrylamide gel media: isoelectric focusing as the first dimension and a separation by molecular size as the second dimension. A two-dimensional pattern of spots each representing a protein is the result of that process. Eventually, spots are made visible by staining or radiographic methods. By analyzing series of such 2DE images one hopes

[1] A preliminary version of this work was presented at the 13th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA) in San Francisco in January 2002 [9]. Carola Wenk was supported by DFG Grant AL 253/4–3.

[2] Computer Science Department, Gould-Simpson 721, University of Arizona, 1040 E 4th St., Tucson, AZ 85721-0077, USA. {alon,carolaw}@cs.arizona.edu.

[3] Institute of Computer Science, Freie Universität Berlin, Takustraße 9, D-14195 Berlin, Germany. {hoffmann,knauer,kriegel,rote}@inf.fu-berlin.de.

Received May 1, 2002; revised January 21, 2003, and June 4, 2003. Communicated by R. C Veltkamp.
Online publication October 24, 2003.

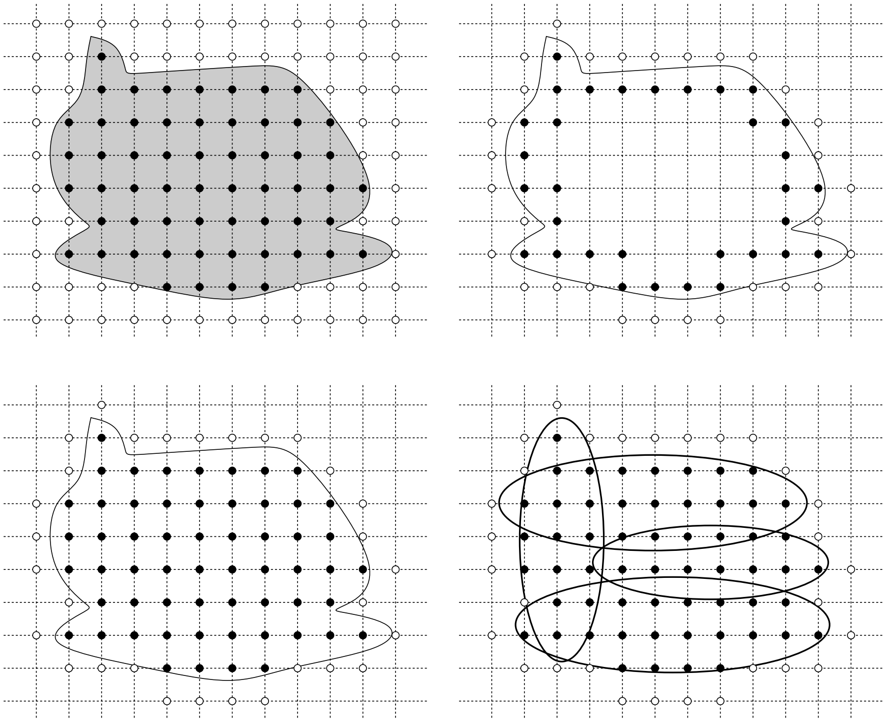

Fig. 1. An instance of the ellipse covering problem. There are various ways how a point set can be derived from a region. We challenge the reader to find a cover with only three ellipses.

to identify those proteins that change their expression (size, intensity) and reflect/cause certain biochemical and biomedical conditions of an organism, see [21]. The first step of the gel analysis, the so-called spot detection, is the algorithmic problem to compute for a given digital gel image all its protein spots. See Figure 2 for an example. Ideally, in a gel image each spot has the shape of an axis-parallel ellipse, which is a widely accepted modeling assumption, see, e.g., [2] and [11].

At first sight spot detection seems to be a pure image processing problem. Usually, one starts with standard techniques like smoothing, segmentation, and background extraction. The resulting image regions correspond ideally to single spots. However, spots that are very close to each other can partially merge (their elliptic shapes overlap) and form rather complicated regions as depicted in Figure 3.

Since in such situations the overlapping spots are often oversaturated (black) the standard image processing methods do not help. In order to solve this problem some heuristics have been implemented in several software packages. However, even then the really complex regions are usually left to be subdivided manually. Our approach is the first attempt to model and solve this problem by means of computational geometry in the following form: *Cover a given planar region $\mathcal{R}$ by the union of a minimal number of axis-parallel ellipses*. As in many applied research problems there are some additional restrictions on the solution coming from the application background. In [10] we have

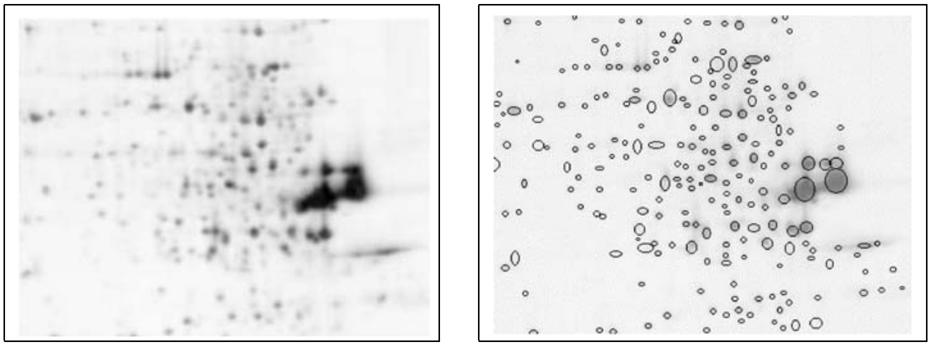

Fig. 2. Part of a gel image and spots computed.

considered an application-specific model of the problem as well as several algorithms for this setting.

Since the data are given as a grid of pixels, it makes sense to discretize the problem and represent the region $\mathcal{R}$ by two sets of points F and R, where R is a sample of required points to be covered (inside $\mathcal{R}$) and F is a set of forbidden points (outside $\mathcal{R}$). The most straightforward way of selecting these sets is to partition all grid points in a suitable bounding rectangle into R and F, as shown in the upper left part of Figure 1.

The set F can be reduced to all pixels outside $\mathcal{R}$ that are adjacent to a point of R on the grid without changing the problem, as shown in the lower left part of Figure 1. Since every connected horizontal or vertical sequence of points of R is now bounded from both sides by a point of F, it follows that no axis-parallel ellipse that covers a point of R can cover a point outside $\mathcal{R}$. (This follows from the fact that the set of grid points in an axis-parallel ellipse is a connected set in the grid.)

The set R can also be reduced by walking along the boundary of $\mathcal{R}$ and choosing points inside within a small distance, as in the upper right part of Figure 1. This approach mimics the general practice of experts who are looking for ellipses approximating long parts of the boundary of $\mathcal{R}$. However, this means essentially that only the boundary of $\mathcal{R}$ is considered, and the computed cover could leave holes in the interior of the region.

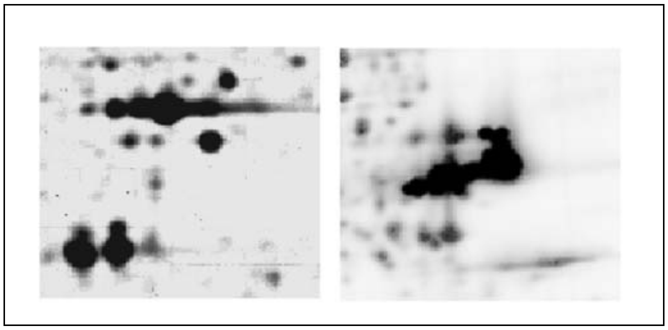

Fig. 3. Twin spots, streaks, and complex region.

Note that the possibility of holes cannot be excluded altogether, as demonstrated by the cover in the lower right part of Figure 1, but this is inherent in the modeling of the problem by a discrete point set.

So we will stick with the choice of R and F that is shown in the lower part of Figure 1. If the spot $\mathcal{R}$ is contained in an $N \times N$ grid, the cardinality $m = |R|$ will typically be quadratic in N, measuring the area of $\mathcal{R}$, whereas F corresponds to the boundary of $\mathcal{R}$ and $n = |F|$ will be only linear in n. So we will typically have $m \approx \text{const} \cdot n^2$. (The relation $m = O(n^2)$ follows from an isoperimetric inequality on the grid graph.) Our algorithm is well adapted to this setting since its running time depends to a higher degree on n than on m.

Related results. The optimization problem of covering a rectilinear polygon with the minimum number of rectangles is NP-hard [8]. This problem is quite similar to our problem of covering a shape with axis-aligned ellipses. This suggests that the ellipse-covering problem might also be NP-hard, although we do not have a proof of this.

Our problem is also related to the problem of covering a shape with strips [1], and to the range covering problem in a hypergraph [5], see [4] for a recent survey on geometric approximation results.

Overview. As every axis-parallel ellipse can be determined by four points, it is easy to see that one can reduce the infinite set of *all* axis-parallel ellipses to a set **S** of size $O((n+m)^4)$ from which the optimal cover is chosen. In this way the problem reduces to a set covering problem. The greedy algorithm for the set covering problem [13] would yield an approximation factor of $O(\log m)$.

We improve this approach in two aspects. Firstly, we replace **S** by a smaller set **C** of so-called canonical objects, which are defined in Section 2. The set **C** contains a cover that is optimal up to a constant factor of 4. We prove subsequently in Section 3 that the size of **C** is only $O(n^2)$ and we describe how to construct it efficiently. The second idea, specified in Section 5, is to adapt the machinery of geometric set cover approximations [5], [7], [15], [20] to select a cover of size $O(k^* \log k^*)$ from **C**, where k^* is the size of the optimal cover. Making use of augmented partition trees, we present an efficient implementation which runs in expected time $\tilde{O}(n^2 + n^{3/2}k^* + mk^* + \sqrt{m}(k^*)^2)$, where $\tilde{O}$ denotes a variant of the O-notation which subsumes polylogarithmic factors. The precise bound is stated in Theorem 3.

We conclude by applying the results to the original gel analysis problem and mentioning a few open problems.

2. Canonical Covers.

We start with two general remarks about the terminology used:

DEGENERATE ELLIPSES. In our proofs we deform an ellipse while keeping some points of its boundary fixed. In this process an ellipse may degenerate into an axis-parallel parabola or even into a hyperplane. Depending on the application, one may or may not permit these "degenerate ellipses" in the covering. We discuss the treatment of these degeneracies in Section 5.2.

CONVENTION. Whenever we speak about ellipses and parabolas, we actually mean *axis-parallel* ellipses and parabolas. A *vertical* or *horizontal* parabola is a parabola with a vertical or horizontal axis, respectively. The size of the optimal cover is denoted by k^*.

Canonical objects. As a first step we show that each ellipse in an optimal cover can be covered by at most four *canonical* objects, each of which is defined by at most four points of F and contains no point of F in its interior. Consequently there exists a cover that uses only canonical objects whose cardinality is at most four times larger than the size of an optimal cover with arbitrary axis-parallel ellipses.

Ideally, we would like the canonical objects to be axis-parallel ellipses that each have at least four points of F on their boundary. However, in general F might be in such a position that additionally we have to consider halfplanes and axis-parallel parabolas, which are degenerate cases of axis-parallel ellipses.

DEFINITION 1. We call an axis-parallel ellipse, an axis-parallel parabola, or a halfplane *F-empty* if it does not contain any point of F in its interior. We call it an *i-point ellipse* (*or parabola or halfplane*) if it is F-empty and additionally contains at least i points of F on its boundary. An i-point ellipse (vertical parabola, horizontal parabola, halfplane) is called *canonical* if there are i points such that it is the only F-empty ellipse (vertical parabola, horizontal parabola, halfplane) with these i points on its boundary.

All 2-point halfplanes and 3-point parabolas are canonical in this sense. In most cases, four points uniquely determine an ellipse, but this is not always true, as for the four corners of an axis-parallel square. Thus, not all 4-point ellipses are canonical. Usually, when we consider an i-point canonical object, a set of i points defining it will be given.

Reduction to canonical objects. The basic idea of the reduction is the following: we pick an axis-parallel ellipse E_0 in an optimal cover; by definition E_0 is F-empty. Now essentially we blow up E_0 to E_0' until it hits a point in F; we continue this process until we have enough points on the boundary of E_0'. During the blow-up we maintain the property that E_0' is F-empty and that it contains E_0. However, in order to maintain this containment property we will have to cover E_0 not by a single ellipse but by up to four ellipses which are derived from E_0.

LEMMA 1. *Let E be an F-empty ellipse. Then there exist E_1, E_2 such that $E \subseteq E_1 \cup E_2$, where E_1, E_2 are either 3-point ellipses or 2-point halfplanes.*

PROOF. We describe a 4-step process that transforms E appropriately: First, scale the plane so that E is a circle. If E does not touch F, increase its radius until a point in F is hit. If E touches only one point p of F, blow it up from p, i.e., move the midpoint m of E away from p on the ray that emanates in m towards p, and increase the radius of E so that it keeps touching p, until it either hits a second point q of F or degenerates to a halfplane. If E becomes a halfplane and still touches only one point of F, rotate two copies E_0, E_1 of E around p in opposite directions, until they both hit a second point; in that case we are finished. Otherwise, if E touches two points p and q of F, move the centers m_0 and m_1 of two copies E_0 and E_1 of E on the bisector of p and q into both

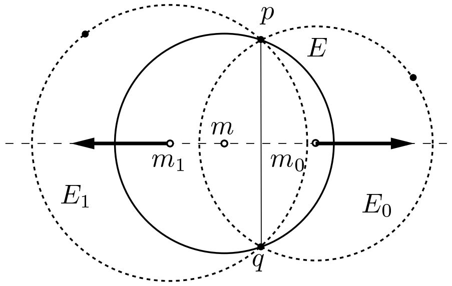

Fig. 4. Illustration of the stopping rule.

directions and keep touching p and q. Continue until each circle either hits a third point of F or degenerates to a halfplane, see Figure 4. □

LEMMA 2. *Let E be a 3-point ellipse. Then there exist E_1, E_2 that have the same three points of F on their boundary, such that $E \subseteq E_1 \cup E_2$, where E_1, E_2 is either a 3-point parabola or a canonical 4-point ellipse.*

PROOF. Assume that $E = \{(x, y) \in \mathbb{R}^2 \mid g(x, y) := ax^2 + by^2 + cx + dy + e \leq 0\}$ with $a + b = 1$ is not already a canonical ellipse. Then there is a 1-parameter family of F-empty ellipses with the same three points as E on their boundary. Let $E \neq E' = \{(x, y) \in \mathbb{R}^2 \mid g'(x, y) := a'x^2 + b'y^2 + c'x + d'y + e' \leq 0\}$ with $a' + b' = 1$ be such an ellipse. Note that $a, b > 0$, $a', b' > 0$, since E and E' are ellipses, and $(a, b) \neq (a', b')$, because otherwise E and E' would intersect at most twice. Thus we can assume without loss of generality that $a > a'$ and $b < b'$. Let $g_\lambda(x, y) := (1 - \lambda)g(x, y) + \lambda g'(x, y)$ and $E(\lambda) := \{ (x, y) \in \mathbb{R}^2 \mid g_\lambda(x, y) \leq 0 \}$.

Now, by the equation $(\lambda - \mu)g(x, y) = \lambda g_\mu(x, y) - \mu g_\lambda(x, y)$, we can conclude that $E \subseteq E(\lambda) \cup E(\mu)$ for all $\mu \leq 0 \leq \lambda$. We let λ grow from zero until at λ_0 either $a_\lambda := a + \lambda(a' - a)$ becomes zero, or a fourth point of F is hit by $E(\lambda_0)$; in the first case, $E_1 := E(\lambda_0)$ is a 3-point horizontal parabola, and in the second case E_1 is a canonical 4-point ellipse. By decreasing μ from zero to μ_0 in a similar way we get $E_2 := E(\mu_0)$. □

COROLLARY 3. *An F-empty ellipse E can be covered by at most four regions which are either 2-point halfplanes, 3-point parabolas, or canonical 4-point ellipses.*

DEFINITION 2. Let $\mathbf{E}^+$, $\mathbf{H}_2$, $\mathbf{P}_3^+$, and $\mathbf{E}_4^+$ denote the set of all F-empty ellipses, the set of all 2-point halfplanes, the set of all 3-point parabolas, and the set of all canonical 4-point ellipses respectively. We call $\mathbf{C} := \mathbf{E}_4^+ \cup \mathbf{P}_3^+ \cup \mathbf{H}_2$ the set of all *canonical objects* for (R, F). A subset $\mathbf{E} \subseteq \mathbf{C}$ with $R \subseteq \operatorname{int}(\bigcup \mathbf{E})$ is called a $\mathbf{C}$-*cover* for (R, F).

COROLLARY 4. *If there is an $\mathbf{E}^+$-cover for (R, F) of size k, then there is a $\mathbf{C}$-cover for (R, F) of size at most $4k$.*

The Delaunay circles of F constitute an F-empty cover of the convex hull of F. By adding halfplanes to cover the exterior of the convex hull, we get an $\mathbf{E}^+$-cover for (R, F) of size less than $2n$. (In fact, this is a cover of $\mathbb{R}^2 - F$.) So we conclude that $k^* \leq \min(m, 2n)$. However, an optimal ellipse cover can be considerably smaller than an optimal circle cover.

3. Constructing the Canonical Objects.

We show that there are only $O(n^2)$ canonical objects, and give an algorithm to construct them all within the same time bound.

4-*Point ellipses*. First we will see how we can construct all 4-point ellipses, and give a quadratic bound on their number by using dynamic Voronoi diagrams.

Let us recall one standard way of constructing the Voronoi diagram of a set F of n points $p = (p_x, p_y)$. The Voronoi diagram can be obtained as the lower envelope of the n bivariate functions $f_p(x, y) := (x - p_x)^2 + (y - p_y)^2$ which measure the squared distance from (x, y) to p. For each point (x, y), the Voronoi cell into which it belongs is determined by the point p with the smallest value $f_p(x, y)$, i.e., the lower envelope of the functions f_p. We can rewrite f_p as follows: $f_p(x, y) := h_p(x, y) + x^2 + y^2$, with $h_p(x, y) = -2p_x x - 2p_y y + p_y^2 w + p_x^2$. Since the expression $x^2 + y^2$ is common to all functions f_p, it plays no role in the computation of the lower envelope, and hence we may as well determine the lower envelope of the n *linear* functions $h_p(x, y)$, i.e., of n planes in 3-space. We use an extended version of this correspondence between Voronoi diagrams and lower envelopes of planes in the proof below.

LEMMA 5. *$\mathbf{E}_4^+$ has at most $\binom{n}{2} - 5$ elements and can be computed in $O(n^2)$ time.*

PROOF. Consider the linear map that maps a point $(x, y) \in \mathbb{R}^2$ to (x, ty) for a parameter $t \in \mathbb{R}$. An F-empty ellipse with width w and height h is, for $t := w/h$, mapped to an $F(t)$-empty circular disk of radius w, where $F(t) := \{(x, ty) \mid (x, y) \in F\}$. So the vertices of the Voronoi diagram of the point set $F(t)$ correspond to $F(t)$-empty disks that have three points of $F(t)$ on their boundary (3-*point disks*), which, after y-scaling by $1/t$ yield F-empty 3-point ellipses. We consider the dynamic Voronoi diagram of $F(t)$, i.e., the Voronoi diagram for varying $t > 0$. Vertices of degree four in this dynamic Voronoi diagram correspond to 4-point disks, which in turn correspond to 4-point ellipses. Regarding time t as a third dimension, this dynamic Voronoi diagram can be considered as the lower envelope of the trivariate distance functions $f_p(x, y, t) := (x-p_x)^2+(y-tp_y)^2$ for $p = (p_x, p_y) \in F$. We can write these functions as $f_p(x, y, t) = h_p(x, ty, t^2) + x^2 + y^2$ with $h_p(u, v, w) = -2p_x u - 2p_y v + p_y^2 w + p_x^2$. The common term $x^2 + y^2$ of all functions f_p can be omitted, and so the vertices of the lower envelope of the original functions f_p correspond to the vertices of the lower envelope of the n hyperplanes h_p in $\mathbb{R}^4$. The lower envelope of n hyperplanes in $\mathbb{R}^4$ has at most $\binom{n}{2} - 5$ vertices and can be computed in $O(n^2)$ time [6], [18]. Therefore, there are only $\binom{n}{2} - 5$ F-empty 4-point ellipses. □

This bound is asymptotically tight in the worst case: Two sets of $n/2$ points on the positive x- and y-axis generate $\Theta(n^2)$ F-empty ellipses. It may however happen that the number of 4-point ellipses is substantially smaller than this bound $O(n^2)$. In this case an alternate procedure may be preferable: The Voronoi diagram of $F(t)$ can be dynamically updated, varying t from 0 to ∞. Each 4-point ellipse corresponds to an event where the combinatorial structure of the Voronoi diagram changes. The update can then be performed in constant time, plus an $O(\log n)$ overhead for maintaining the event queue. An event is triggered when an edge of the Voronoi diagram is reduced to length 0. This procedure constructs $\mathbf{E}_4^+$ in time $O(|\mathbf{E}_4^+|\log n)$.

3-*Point parabolas*. Next we prove that the number of 3-point parabolas is only linear and describe how to compute them in $O(n \log n)$ time.

LEMMA 6. $\mathbf{P}_3^+$ *has size at most* $4n - 10$ *and can be computed in* $O(n \log n)$ *time*.

PROOF. We argue without loss of generality that the number of parabolas with a *vertical* axis is at most $2n - 5$. One way to see this is to look at the limit of the dynamic Voronoi diagram in the proof of Lemma 5 as $t \to 0$. However, we use a more direct argument. We map all the points $p = (x, y) \in F$ to $p' = (x, y, x^2)$; this corresponds to lifting F to a point set F' on the parabolic cylinder ψ given by the equation $z = x^2$. Note that *every* vertical axis-parallel parabola P is the projection of the intersection curve of ψ with an appropriate (unique) plane h_P. Moreover, a point p is contained in P iff p' is below h_P.

This implies that a plane h_P that corresponds to an F-empty axis-parallel parabola P has to lie completely below the lower convex hull of F'. Moreover, a plane that corresponds to a 3-point parabola has to touch this hull in at least three non-collinear points from F'; therefore it corresponds to (i.e., contains) a facet of that hull. The lower convex hull of n points in 3-space has at most $2n - 5$ facets. This shows that there are at most $2n - 5$ such parabolas and we can compute them all in $O(n \log n)$ time by constructing convex hulls in three dimensions. □

2-*Point halfplanes*. The 2-point halfplanes correspond to the edges of the convex hull of F. Thus there are at most n such halfplanes and they can be computed in $O(n \log n)$ time:

LEMMA 7. $\mathbf{H}_2$ *has size at most n and can be computed in* $O(n \log n)$ *time*.

By adding up the numbers from the three previous lemmas, we obtain:

COROLLARY 8. *There are less than* n^2 *canonical objects in* $\mathbf{C}$, *and they can be found in* $O(n^2)$ *time*.

4. The Range Space of Ellipses.

We review basic definitions and results about range spaces, VC-dimension, and ε-nets. The relevant material can be found in [3], [12], [14], and [19].

A *range space* $\mathcal{H} = (V, \mathcal{S})$ consists of a finite ground set V and a finite family $\mathcal{S}$ of subsets of V, the so-called *ranges*. The VC-dimension of a range space $\mathcal{H} = (V, \mathcal{S})$ is the size of the largest subset $A \subseteq V$ such that all subsets $B \subseteq A$ are of the form $A \cap S$

for some $S \in \mathcal{S}$, i.e., $2^A = \{A \cap S \mid S \in \mathcal{S}\}$. A collection of ranges $\{S_B \mid B \subseteq A\}$ with the property $B = A \cap S_B$ is called a shattering set for A.

Instead of the natural choice of taking the ellipses as ranges, we have to consider the dual range space, where the canonical ellipse $\mathbf{C}$ form the ground set V, and the ranges $\mathcal{S}$ correspond to the required points R. More precisely, the range set $\mathcal{S} = \mathcal{S}_R$ consists of the subsets $C_r \subseteq \mathbf{C}$ that contain a required point $r \in R$ in their interior, i.e., $\mathcal{S}_R = \{C_r \mid r \in R\}$ with $C_r = \{E \in \mathbf{C} \mid r \in \text{int}(E)\}$.

Thus, the VC-dimension of $(\mathbf{C}, \mathcal{S}_R)$ is the size of the largest set A of canonical objects, such that for any subset $B \subseteq A$ there is a point $r_B \in R$ which is in $\bigcap_{E \in B} \text{int}(E) \setminus \bigcup_{E \in A \setminus B} \text{int}(E)$.

LEMMA 9. *The VC-dimension of the range space $\mathcal{H} = (\mathbf{C}, \mathcal{S}_R)$ is at most $d_{(\mathbf{C},\mathcal{S}_R)} := 4$.*

PROOF. Suppose that a set A consisting of five objects is shattered by the ranges corresponding to a set $P \subseteq R$ of $2^{|A|} = 32$ required points.

There must be a point $p_0 \in P$ which is not covered by any object in A, i.e., $p_0 \in \mathbb{R}^2 \setminus \bigcup_{E \in A} \text{int}(E)$. By translating the points and ellipses, we can assume that p_0 is the origin. Thus, the interior of any $E \in A$ can be written in the form $\text{int}(E) = \{(x, y) \mid ax^2 + bx + cy^2 + dy + e < 0\}$ with $e \geq 0$, including the case of parabolas and halfplanes. Since P is finite, each e can be increased by some small amount to $e' > 0$ (shrinking E to E') such that $p \in \text{int}(E) \iff p \in \text{Int}(E')$ for all $p \in P$ and $E \in A$. Now, a point $p \in P$ is either in the interior or in the exterior of an object E', but never on the boundary. Lifting the points $p = (p_x, p_y) \in P$ to $l(p) = (p_x^2, p_x, p_y^2, p_y)$ we observe that $p \in \text{int}(E)$ implies that $l(p)$ is in the open halfspace $H_E^- = \{x \in \mathbb{R}^4 \mid ax_1 + bx_2 + cx_3 + dx_4 + e' < 0\}$, whereas $p \notin \text{int}(E)$ implies that $l(p) \in H_E^+ = \{x \in \mathbb{R}^4 \mid ax_1 + bx_2 + cx_3 + dx_4 + e' > 0\}$. This way the assumption that A is shattered by P is equivalent to the condition that the points $\{l(p) \mid p \in P\}$ are in different 4-cells an arrangement of five hyperplanes in $\mathbb{R}^4$. This is a contradiction because any arrangement of $d + 1$ hyperplanes in $\mathbb{R}^d$ has at most $2^{d+1} - 1$ d-cells. Thus, the VC-dimension of $(\mathbf{C}, \mathcal{S}_R)$ is at most 4. □

Let $w : V \to \mathbb{N}$ be a weight function on the vertex set of the range space $\mathcal{H} = (V, \mathcal{S})$. The weight of a subset $X \subseteq V$ is defined as $w(X) := \sum_{v \in X} w(v)$. For $\varepsilon > 0$, a set $\mathbf{E} \subseteq V$ is called an *ε-net* for the range space $\mathcal{H}$ with respect to the weight function w if $\mathbf{E} \cap S \neq \emptyset$, for any $S \in \mathcal{S}$ with $w(S)/w(V) > \varepsilon$.

LEMMA 10 [16]. *Let $\mathcal{H} = (V, \mathcal{S})$ be a range space of VC-dimension d, let $w : V \to \mathbb{N}$ be a weight function on V, and let $\varepsilon > 0$. Then a set of* $(d/\varepsilon)(\log(1/\varepsilon) + 2 \log\log(1/\varepsilon) + 3)$ *points from V drawn independently* (*i.e., with possible repetition*) *according to the weight function w, constitutes an ε-net for the range space $\mathcal{H}$ with respect to the weight function w with probability at least* 1/2.

5. The Covering Algorithm. We describe a randomized algorithm that computes a $\mathbf{C}$-cover for (R, F) which consists of $O(k^* \log k^*)$ canonical objects. The main techniques applied here have been developed in [5], [7], [15], and [20]. For each ellipse $E \in \mathbf{C}$ there is a weight $w(E)$ which estimates the importance of E for covering R. The weight of a set $V \subseteq \mathbf{C}$ is given by $w(V) := \sum_{E \in V} w(E)$. The algorithm proceeds in rounds. We start with a simple version of the algorithm, which will be successively refined. In what follows $c := 2d_{(\mathbf{C},\mathcal{S}_R)} = 8$.

Algorithm 1

Input: (R, F) and a parameter $k > 0$.
Output: If the algorithm terminates, it returns a $\mathbf{C}$-cover $\mathbf{E}$ for (R, F) of size $|\mathbf{E}| \le ck \log k$.

```
1. Initially set $w(E) = 1$ for all $E \in \mathbf{C}$.
2. Start a new *round* by picking a random sample $\mathbf{E}$ of size $ck \log k$ from $\mathbf{C}$
   according to the weight distribution $w$.
3. If $\mathbf{E}$ is a cover, halt.
4. Take a point $q \in R$ which is not covered by $\mathbf{E}$, and determine the set
   $C_q = \{E \in \mathbf{C} \mid q \in E\}$.
5. If $w(C_q) \le w(\mathbf{C})/(2k)$, then this round is declared to be *successful* and the
   weight of all $E \in C_q$ is doubled.
6. Goto Step 2.
```

Lemma 11. *If $k \ge 4k^*$, then*

1. *If the algorithm does not halt in a round, the probability that the round is successful is at least* $1/2$, *and*
2. *the number of successful rounds is at most* $16k^* \log(n^2/k^*) \le 8k \log n$.

Proof. 1. Let $\varepsilon := 1/(2k)$ and consider the range space $\mathcal{H} = (\mathbf{C}, \mathcal{S}_R)$. Recall that $\mathcal{S}_R = \{C_r \mid r \in R\}$ and $C_r = \{E \in \mathbf{C} \mid r \in E\}$. From Lemmas 9 and 10 we can conclude that a random sample $\mathbf{E}$ of size $ck \log k$ from $\mathbf{C}$ is an ε-net for $\mathcal{H}$ with respect to the weight function w with probability at least $1/2$. Thus for any $X \subseteq \mathbf{C}$ with $w(X) \ge \varepsilon w(\mathbf{C})$ it follows that $\mathbf{E} \cap X \ne \emptyset$. Now if $\mathbf{E}$ is indeed an ε-net, and $q \in R$ is not covered, i.e., $\mathbf{E} \cap C_q = \emptyset$, it follows that $w(C_q) \le \varepsilon w(\mathbf{C})$, so the round is successful.

2. In each successful round the total weight $w(\mathbf{C})$ increases by a factor of at most $(1 + \varepsilon) \le e^{\varepsilon} \le 2^{3/(4k)} \le 2^{3/(16k^*)}$. Thus, after s successful rounds $w(\mathbf{C}) \le n^2 2^{3s/(16k^*)}$, using the fact that $|\mathbf{C}| < n^2$ (Corollary 8). Let $\mathbf{E}_0$ be an optimal $\mathbf{C}$-cover. By Corollary 4, we know that $k^* \le |\mathbf{E}_0| \le 4k^*$. Since $\mathbf{E}_0$ covers R, $\mathbf{E}_0 \cap C_q \ne \emptyset$ in each round, so in each successful round the weight of at least one $E \in \mathbf{E}_0$ is doubled. Now if d_E denotes the number of times that the weight of $E \in \mathbf{E}_0$ has been doubled after s successful rounds, then $\sum_{E \in \mathbf{E}_0} d_E \ge s$, and we can conclude $w(\mathbf{E}_0) = \sum_{E \in \mathbf{E}_0} 2^{d_E} \ge |\mathbf{E}_0| 2^{s/|\mathbf{E}_0|} \ge k^* 2^{s/(4k^*)}$, where the penultimate inequality follows from Jensen's inequality. Since $w(\mathbf{E}_0) \le w(\mathbf{C})$ we finally get $s \le 16k^* \log(n^2/k^*)$. □

If $k \ge 4k^*$ we can view a single round of Algorithm 1 as a Bernoulli experiment with success probability at least $1/2$. We consider the probabilty of the event that Algorithm 1 does not halt after $8tk \log n$ rounds (implying that the number of successful rounds was smaller than $8k \log n$). This is bounded by the probability that the sum S of $8tk \log n$

independent random variables X_i with $Pr[X_i = 1] = Pr[X_i = -1] = \frac{1}{2}$ is larger than $\lambda = 8(t-2)k \log n$. Applying Chernoff bounds we obtain the following inequality:

$$Pr[S > \lambda] \le e^{-\lambda^2/(2\cdot 8k \log n)} = e^{-((t-2)^2/t)\ 4k \log n}.$$

For $t \ge 8$ we have $(t-2)^2/t \ge 2t$ and $Pr[S > \lambda] \le e^{-8tk \log n}$. Now consider the following algorithm (call it **Algorithm 2**): Given k and $\delta > 0$, we run Algorithm 1 for up to $\max(64k \log n, \ln(1/\delta))$ rounds. This way the number of rounds is $8tk \log n$ with $t \ge 8$ and $t \ge \ln(1/\delta)/(8k \log n)$. By the estimations above, the algorithms stops with a cover of size at most $8k \log k$ with probability at least $1 - \delta$. Otherwise we halt after $\max(64k \log n, \ln(1/\delta))$ rounds. This constitutes a randomized approximation algorithm for the decision problem variant of the minimal cover problem with a one-sided error:

THEOREM 1. *Given k and $\delta > 0$, Algorithm 2 stops after $\max(64k \log n, \ln(1/\delta))$ rounds, and if $k \ge 4k^*$ it returns a cover of size at most $8k \log k$ with probability at least $1 - \delta$.*

Since the value of k^* is not known beforehand, we have to perform an exponential search for it: We run Algorithm 2 for $k = 2, 4, 8, 16, \ldots$ until it finds a cover (call this procedure **Algorithm 3**). In the $\lceil \log k^* \rceil$th step of the exponential search (when $4k^* \le k \le 8k^*$), the algorithm is successful with probability at least $1 - \delta$, and we get a cover of size at most $8ck^* \log(8k^*)$. The total runtime of the exponential search procedure is dominated by the runtime of the last step.

THEOREM 2. *For any $\delta > 0$, Algorithm 3 computes after $O(k^* \log(n) \log(1/\delta))$ rounds a cover of size at most $64k^* \log 8k^*$ with probability at least $1 - \delta$.*

5.1. *Data Structures and Algorithms for the Individual Steps.* It remains to devise efficient means and data structures to maintain the weights of the objects in **C** such that they allow efficient sampling according to w. Moreover, we have to specify how to check whether a candidate sample **E** constitutes a cover. We first assume that **C** does not contain parabolas or halfplanes. Below we show how to modify our algorithm to handle these objects as well.

Maintaining the weights. Each of the $O(n^2)$ ellipses $E \in \mathbf{C}$ is specified by four real parameters and can be written in the following form:

$$E = \{(x, y) \in \mathbb{R}^2 \mid g(x, y) := a(x^2 - y^2) + bx + cy + d + y^2 \le 0\}, \tag{1}$$

where $0 < a < 1$ and $b, c, d \in \mathbb{R}$. We now perform a similar transformation as in the proof of Lemma 9, except that we have used a different normalization for the equation of the ellipse. The ellipse E contains a point $p = (x, y)$ iff $g(x, y) \le 0$. If we map E to the point $p_E := (a, b, c, d) \in \mathbb{R}^4$ and the point p to the hyperplane $h_p := \{(A, B, C, D) \in \mathbb{R}^4 \mid A(x^2 - y^2) + Bx + Cy + D + y^2 = 0\}$, then E contains p iff p_E is below h_p.

We identify each ellipse $E \in \mathbf{C}$ with the point p_E. Let $\mathbf{C}'$ be the set of these points. In order to pick an ellipse at random efficiently and to maintain the weights efficiently we store $\mathbf{C}'$ in a partition tree data structure: The partition tree of [17] for these $N = O(n^2)$ points can be constructed in $O(N \log N) = O(n^2 \log n)$ time, $O(N)$ space, and allows halfspace range queries to be answered in time $O(N^{3/4} \log^{O(1)} N) = O(n^{3/2} \log^{O(1)} n)$. The first level of the tree stores a partition of $\mathbb{R}^4$ into $O(N^{3/4})$ simplices, where each simplex contains $N^{1/4}$ points. Recursively, a simplex representing r points stores a simplicial partition of size $O(r^{3/4})$. The height of the tree is $O(\log \log N)$. In this tree data structure the points themselves are stored only at the leaves. For our purposes we add the weight information for the points to the tree as follows: We store at each node a factor, initially set to 1. The weight for an ellipse (in a leaf) is the product of the factors on the path from the leaf to the root.

Now suppose an uncovered point $q \in R$ is given, for which we need to double the weights of all ellipses in $\mathbf{C}$ that contain q. In other words, we have to double the weights of all points in $\mathbf{C}'$ that lie above h_q. This can be done using the halfspace range query algorithm of [17] which touches all simplices in the partition, and then goes recursively into those simplices that are cut by h_q. When touching all simplices in a level, we simply have to double the factors of those simplices that are completely above h_q. So the doubling of the weights can be done in $O(n^{3/2} \log^{O(1)} n)$ time.

Random sampling. In order to pick an ellipse at random efficiently from the tree we have to add additional information to each node: In every inner node v we store the sum s_v of all weights in the subtree rooted at s_v, divided by all factors on the path from v (not including v) to the root. Note that we can initialize all s_v easily in a bottom-up manner. To find a random sample we recursively go down the tree from the root, picking a random child at each vertex, according to the weights that are stored in the children. To this end we store in each node a sorted list of adjacent intervals whose lengths are the weights of the children. In order to go to a random child, we generate a random number in the union of the intervals and find the interval containing it by binary search in $O(\log n)$ time. We can now pick a random ellipse by following a random path from the root to a leaf in $O(\log n \log \log n)$ time. According to Lemma 10 the sample $\mathbf{E}$ can be generated by independent draws of random ellipses, with replacement.

During a weight doubling step we can maintain the interval partitions at asymptotically no extra cost since during a query we touch all children of each node that we visit in the recursion anyway.

Verifying the cover. Now we need to check if $\mathbf{E}$ covers R. We first give a simple algorithm which we speed up afterwards with a batching technique. We proceed as follows: Compute the arrangement of the $k_1 := ck \log k$ ellipses, together with an efficient point location data structure in $O(k_1^2 \log k_1)$ time; then query this data structure with all points in R. This takes $O(m \log k_1)$ time and identifies an uncovered point. Now if $k_1 \le \sqrt{m}$ the total time spent in that procedure is $O((m + k_1^2) \log k_1) = O(m \log k_1) = O(m \log m)$. If $k_1 > \sqrt{m}$ we can split $\mathbf{E}$ into $g := \lceil k_1/\sqrt{m} \rceil$ groups of size at most $\sqrt{m}$ and run the previously described procedure for each of these groups. This requires $O(k_1 \sqrt{m} \log m)$ time. To summarize, we can identify an uncovered point $q \in R \setminus \bigcup \mathbf{E}$ in $O((m + k_1\sqrt{m}) \log m) = O((\sqrt{m} + k \log k)\sqrt{m} \log m)$ time.

Putting all this together, Algorithm 3 needs:

1. $O(n^2 \log n)$ preprocessing time to initialize the partition tree, and

2. in each of the $O(k^* \log(n) + \log(1/\delta))$ rounds
   (a) $O(n^{3/2} \log^{O(1)} n)$ time for the weight update and the sampling step, and
   (b) $O((\sqrt{m} + k^* \log k^*)\sqrt{m} \log m)$ time for the verification step.

THEOREM 3. *For any* $\delta > 0$, *Algorithm* 3 *computes with probability at least* $1 - \delta$ *in* $O(n^2 \log n + (k^* \log(n) + \log(1/\delta))[(n^{3/2} \log^{O(1)} n + (\sqrt{m} + k^* \log k^*)\sqrt{m} \log m)]) = \tilde{O}(n^2 + k^* n^{3/2} + k^* m + (k^*)^2\sqrt{m})$ *time a cover of size at most* $2ck^* \log k^*$.

It may happen that the number of 4-point ellipses is substantially smaller than the bound $O(n^2)$. This translates directly into a corresponding improvement of the running time to $\tilde{O}(|\mathbf{E}_4^+| + k^*|\mathbf{E}_4^+|^{3/4} + k^* m + (k^*)^2\sqrt{m})$, see the remark after Lemma 5.

5.2. *Handling Degenerate Cases.* To finish the description of our approximation algorithm we need to clarify a few points. First we have to show how to adapt our method so that it can handle axis-parallel parabolas and halfplanes. Next, since our ultimate goal is to find a cover with ellipses only, we also have to describe how to repair a cover computed by the algorithm so that it only uses ellipses. This is actually quite straightforward in the original setting but if we relax the covering condition to allow covered points on the boundary of covering objects, this issue gets slightly more intricate.

Parabolas and halfplanes. First note that axis-parallel parabolas can also be written in the form of (1) if we allow $0 \leq a \leq 1$. Therefore the algorithm we just described can handle them without any modifications.

For halfplanes, we can adapt the techniques that work for parabolas and ellipses. Each halfplane is represented by a point in a dual space, which is just two-dimensional in this case. In order to find all halfplanes that contain a point $q \in R$, we have to perform a halfplane range-query in the dual setting. We can also use efficient data structures for this problem and augment them appropriately with the weight information for the halfplanes. Thus we end up with two data structures: one that handles ellipses and parabolas and one that handles halfplanes. In the sampling step we first decide, depending on the total weight of the data structures, whether to take a halfplane or an ellipse/parabola, and then continue the sampling in the appropriate data structures as described above. The asymptotic performance of the algorithm is not affected by this modification.

Since the covering relation is strict, i.e., no point of R lies only on boundaries of canonical objects, each halfplane and parabola in a $\mathbf{C}$-cover can in the end be perturbed into an F-empty ellipse covering the same points of R. This can be done in $O(m)$ time for each halfplane or parabola. However, the following approach avoids this overhead: We select for each point of R only one object that is required to cover it. (This can be carried out during the check that a covering is given, by the algorithm in Section 5.1.) The final conversion can then be done in $O(K + m)$ time for a total of K parabolas and halfplanes in the cover, since different parabolas and halfplanes have to take care of disjoint sets of R-points.

Non-strict covers. We can modify our approach so that it also works when we allow the points of R to be covered by the boundary of the covering objects. We call a set of axis-parallel ellipses $\mathbf{E}$ a *non-strict* cover of (R, F) if the union $\bigcup \mathbf{E} := \bigcup_{E \in \mathbf{E}} E$ covers R and respects F, i.e., $R \subseteq \bigcup \mathbf{E}$ and $F \cap \operatorname{int}(\bigcup \mathbf{E}) = \emptyset$. All our previous arguments and

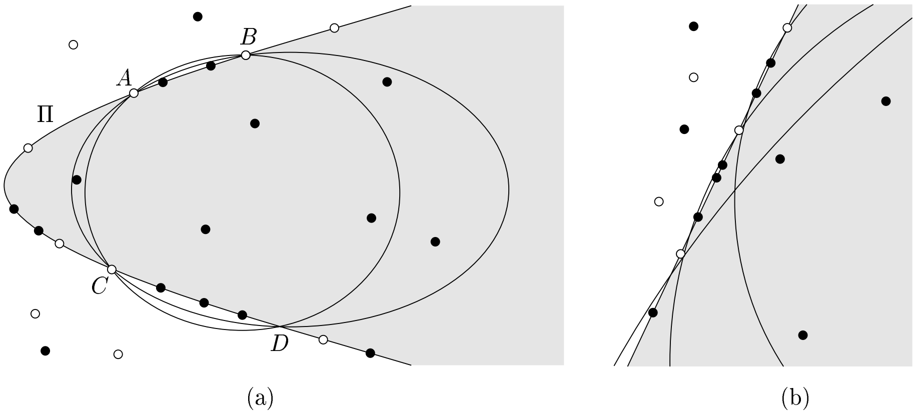

Fig. 5. A parabola and a hyperplane in a non-strict covering.

algorithms carry over to this setting. In particular, we can compute a non-strict **C**-cover for (R, F) of size $O(k^* \log k^*)$ within the time bounds stated in Theorem 3.

The only difficulty arises in the last step when we have to replace halfplanes and parabolas by ellipses. Consider the parabola shown in Figure 5(a). Clearly, a true F-empty ellipse cannot cover the same set of R-points as this parabola, because any ellipse can intersect the parabola in at most four points, and hence it can cut out at most two intervals of the parabola. Thus, if we want to insist on real ellipses and want to exclude parabolas and halfplanes, we have to modify the construction of the candidate set **C**:

By the arguments of Section 2, the parabolas that we have to consider arise as limits of a family of 3-point ellipses (see Lemma 2). Consider the dynamic Voronoi diagram in the proof of Lemma 5. We increase the scaling parameter t beyond the point where the last combinatorial change in the Voronoi diagram occurs. In other words, we select t larger than the largest value that corresponds to a 4-point ellipse. Each vertex in the Voronoi diagram of $F(t)$ represents now a 3-point ellipse which "converges" to a horizontal parabola with the same three points on the boundary. The vertical parabolas can be found analogously by choosing t close enough to zero.

There are $O(n)$ of these ellipses. Suppose that such a 3-point ellipse is given by three points A, B, C, see Figure 5(a). All ellipses through A, B, C intersect the 3-point parabola Π given by A, B, C in the same fourth point D. Thus, the set of points *on* Π that can be covered is fixed. It is now easy to select an ellipse close enough to Π such that all points of R in the interior of Π are covered as well. This would take $O(m)$ time, for a total time of $O(mn)$. Alternatively, it is possible to represent the ellipse symbolically, by storing the parabola Π and the points A, B, C, D. Then it can be decided in $O(1)$ time whether a given point of R is covered by the ellipse. The conversion to true ellipses can be done at the end for the parabolas that are selected for the covering, as described above for the case of strict covering.

Halfplanes are much simpler to deal with, see Figure 5(b): we just create an ellipse (it can even be a circle) for each set of R-points on the boundary which form an interval which is not interrupted by F-points.

5.3. *The Application Revisited.* In the spot detection application for electrophoresis gels which we have described in Section 1 the task is to cover a planar region by the union of a minimal number of axis-parallel ellipses. Since for the computer-assisted analysis the electrophoresis gels are scanned, the planar region is given as a pixel pattern. As was argued in the Introduction, $n = |F|$ is approximately the length of the boundary of the spot in this setting, and $m = |R| = O(n^2)$. Since every connected horizontal or vertical sequence of points of R is bounded from both sides by a point of F, halfplanes or parabolas cannot occur in a cover, so we need not take the trouble to handle these special cases. By Theorem 3, we obtain a cover of size $O(k^* \log k^*)$ cover in expected time $\tilde{O}(n^2 k^*)$. We can even omit the partition tree data structure and find uncovered points and update the weights trivially in $O(n^2)$ time and still achieve this time bound.

5.4. *Open Questions.* Whether our ellipse-covering problem is really NP-hard is of course an interesting open problem. The proof of [8] for rectangle coverings is quite involved, and it will be technically very difficult to extend it to ellipses.

Our problem has its origin in image processing. We have discretized the problem by choosing points sets R and F as "hard" constraints for the ellipses. Other approaches are conceivable. One could specify some maximum number of ellipses and minimize the number of uncovered points of R, of covered points of F, or some combination. Each pixel might have a different propensity to be covered or uncovered, based on its gray-level, and one might minimize some objective function based on the "mis-classified" pixels.

One could also model the geometry differently, by specifying some tolerance and defining an annular region around the boundary of the spot in the image, limited by an "inner boundary" and an "outer boundary." One looks for a set of ellipses whose union covers the inner boundary including the interior but remains within the outer boundary. Even if these boundaries are polygonal, our approach does not readily generalize to this setting because it depends crucially on the finiteness of the sets F and R.

Acknowledgments. We thank Helmut Alt, Sariel Har-Peled, and Ulrich Kortenkamp for fruitful discussions.

References

[1] P. K. Agarwal and C. M. Procopiuc. Approximation algorithms for projective clustering. In *Proc.* 11*th ACM–SIAM Symposium on Discrete Algorithms*, pages 538–547, 2000.

[2] R. Appel, J. Vargas, P. Palagi, D. Walther, and D. Hochstrasser. Melanie II, a third-generation software package for analysis of two-dimensional electrophoresis images: II. Algorithms. *Electrophoresis*, 18:2735–2748, 1997.

[3] P. Assouad. Densité et dimension. *Ann. Inst. Fourier* (*Grenoble*), 3:232–282, 1983.

[4] M. Bern and D. Eppstein. Approximation algorithms for geometric problems. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, pages 296–345. PWS, Boston, MA, 1997.

[5] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.*, 14:263–279, 1995.

[6] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.

[7] K. L. Clarkson. Algorithms for polytope covering and approximation. In *Proc*. 3*rd Workshop on Algorithms and Data Structures*, volume 709 of Lecture Notes in Computer Science, pages 246–252. Springer-Verlag, Berlin, 1993.

[8] J. Culberson and R. A. Reckhow. Covering polygons is hard. *J. Algorithms*, 17:2–44, 1994.

[9] A. Efrat, F. Hoffmann, C. Knauer, K. Kriegel, G. Rote, and C. Wenk. Covering shapes by ellipses. In *Proc*. 13*th Annual ACM–SIAM Symposium on Discrete Algorithms* (*SODA*), pages 453–454, San Francisco, January 2002.

[10] A. Efrat, F. Hoffmann, K. Kriegel, C. Schultz, and C. Wenk. Geometric algorithms for the analysis of 2d-electrophoresis gels. In *Proc*. *Fifth Annual International Conference on Computational Molecular Biology* (*RECOMB*), pages 114–123, Montréal, 2001.

[11] J. Garrels. The QUEST system for quantitative analysis of 2D gels. *J. Biol*. *Chem*., 264:5269–5282, 1989.

[12] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput*. *Geom*., 2:127–151, 1987.

[13] D. S. Hochbaum. Approximation algorithms of the set covering and vertex cover problems. *SIAM J. Comput*., 11(3):555–556, 1982.

[14] J. Komlós, J. Pach, and G. Woeginger. Almost tight bounds for ε-nets. *Discrete Comput*. *Geom*., 7:163–173, 1992.

[15] N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. In *Proc*. 28*th Annual IEEE Symposium on Foundations of Computer Science*, pages 68–77, 1987.

[16] J. Matoušek. Cutting hyperplane arrangements. *Discrete Comput*. *Geom*., 6:385–406, 1991.

[17] J. Matoušek. Efficient partition trees. *Discrete Comput*. *Geom*., 8:315–334, 1992.

[18] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete Comput*. *Geom*., 6:423–434, 1991.

[19] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl*., 16:264–280, 1971.

[20] E. Welzl. Partition trees for triangle counting and other range searching problems. In *Proc*. 4*th Annual ACM Symposium on Computational Geometry*, pages 23–33, 1988.

[21] M. R. Wilkins, K. L. Williams, R. D. Appel, and D. F. Hochstrasser, editors. *Proteome Research*: *New Frontiers in Functional Genomics*. Springer-Verlag, New York, 1997.