Covering with Ellipses

Alon Efrat[†], Frank Hoffmann[‡], Christian Knauer[‡],
Klaus Kriegel[‡], Günter Rote[‡], Carola Wenk[‡,⋆]

[†]University of Arizona, Computer Science Department,
Gould-Simpson 721, P.O. Box 210077, Tucson, Arizona, U.S.A.
alon@CS.Arizona.EDU
[‡]Freie Universität Berlin, Institute of Computer Science,
Takustrasse 9, D-14195 Berlin, Germany,
{hoffmann,knauer,kriegel,rote,wenk}@inf.fu-berlin.de

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.

Keywords: 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 $\mathbf{E} = \{E_1, \ldots, E_{k_0}\}$ of axis-parallel ellipses, of minimal cardinality k_0, such that their union $\cup\mathbf{E} := \cup_{E\in\mathbf{E}} E$ *covers* R and *strictly respects* F, i.e., $R \subseteq \mathrm{int}(\cup\mathbf{E})$ and $F \cap \mathrm{int}(\cup\mathbf{E}) = \emptyset$. Thus $\cup\mathbf{E}$ has to fully contain R in its interior and may contain no points from F except on its boundary.

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. We challenge the reader to find a cover with only three ellipses.

⋆ Supported by DFG grant AL 253/4–3.

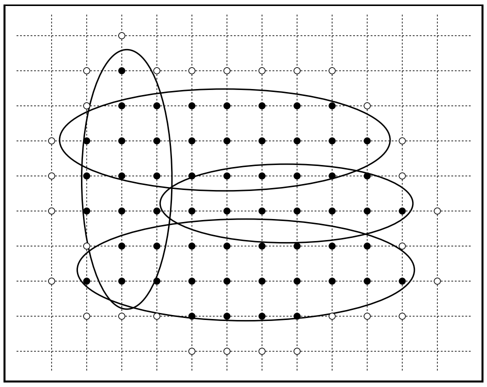


Fig. 1. An instance of the ellipse covering problem

Motivation. This problem stems from a pattern recognition task in proteomics, which is a rapidly growing field within molecular biology. In proteomics two–dimensional gel 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 to identify those proteins that change their expression (size, intensity) and reflect/cause certain biochemical and biomedical conditions of an organism, see [15]. 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, 7].

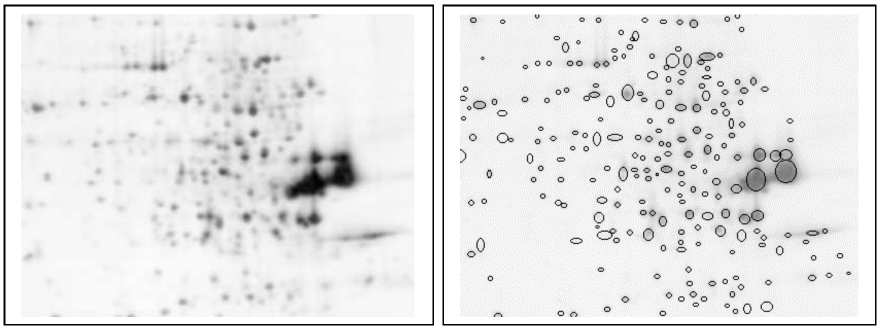


Fig. 2. Part of a gel image and spots computed

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.

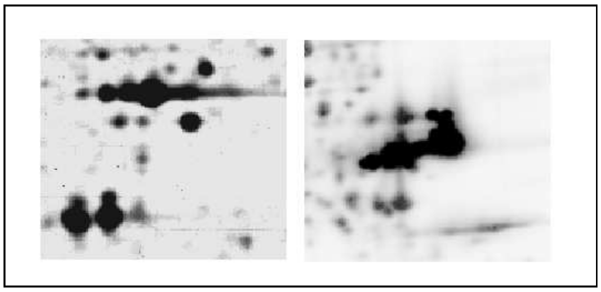


Fig. 3. Twin spots, streaks and complex region

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. But even then, the really complex regions are usually left to be subdivided by the user. 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 [6] we have considered an application specific model of the problem as well as several algorithms for this setting.

Related results. From the theoretical point of view the optimization problem of covering a shape with ellipses (with small Hausdorff distance) is closely related to the problem of exactly covering a shape with rectangles, which was shown to be NP-complete [5]. It is also related to the problem of covering a shape with strips [1], and to the range covering problem in a hypergraph [3]. Thus, in the general setting there is not much hope for finding a polynomial time algorithm. Consequently, we are looking for approximation algorithms.

To make use of the powerful machinery of geometric approximation algorithms the region $\mathcal{R}$ will be represented 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 R). One can obtain these sets walking along the boundary of $\mathcal{R}$ and choosing points inside and outside within a small distance. This approach somehow mimics the general practice of experts who are looking for ellipses approximating long parts of the boundary of $\mathcal{R}$. The advantage that both sets are of small cardinality has to be paid for by the fact that the computed cover could have some hole in the interior of the region. To avoid this one also can choose

R as the set of all grid points in $\mathcal{R}$ (from an appropriately dense grid). Then, of course, the cardinality m of R can be quadratic in n, the size of F.

Overview. The naive approach to the general ellipse covering problem would apply the greedy algorithm for the set covering problem [9] to a set $\mathbf{S}$ of ellipses, which contains a cover of optimal size k_0. It is easy to see that there is such a set $\mathbf{S}$ of size $O((n+m)^4)$. This yields an approximation factor of $O(\log m)$.

Our alternative approach to solve the problem affects both sides of the greedy solution. Firstly, we substitute $\mathbf{S}$ by a smaller set $\mathbf{C}$ of so-called canonical objects. In Section 2 we create such a set $\mathbf{C}$ and show that it contains a cover that is optimal up to a constant factor. We prove subsequently in Section 3 that the size of $\mathbf{C}$ is only $O(n^2)$ and we describe how to construct it efficiently. The second idea, specified in Section 4, is to adapt the machinery of geometric set cover approximations [10, 14, 4, 3] to select a $O(k_0 \log k_0)$ cover from $\mathbf{C}$. Making use of augmented partition trees, we present an efficient implementation which runs in expected time $\tilde{O}(n^2 + n^{3/2}k_0 + mk_0 + \sqrt{m}k_0^2)$, where $\tilde{O}$ denotes a variant of the O-notation which subsumes polylogarithmic factors. We conclude with applying the results to the original gel analysis problem.

Convention: Whenever we speak about ellipses and parabolas, we actually mean *axis-parallel* ellipses and parabolas.

2 Canonical covers

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.

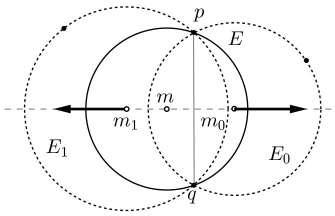
Definition 1. *We call an axis-parallel ellipse, an axis-parallel parabola, or a halfplane* F-empty *if it does not contain any point of F in its interior. We call it an i-*point ellipse (or parabola or halfplane) *if it is F-empty and additionally contains at least i points of F on its boundary. An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

All 2-point halfplanes and 3-point parabolas are canonical in this sense. In general, four points uniquely determine an ellipse, but not all 4-point ellipses are canonical, as four points at the corners of a square show.

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 only two points 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 directions and keep touching p and q. Continue until each circle either hits a third point of F or degenerates to a halfplane. $\square$



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 one-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 $(a,b) \neq (a',b')$, since E and E' are not homothetic. Thus we can assume wlog. that $a > a'$ and $b < b'$. Let $E(z) := \{(x,y) \in \mathbb{R}^2 \mid g_z(x,y) := (1-z)g(x,y) + zg'(x,y) \leq 0\}$.

Now, since $(\lambda - \mu)g(x,y) = \lambda g_\mu(x,y) - \mu g_\lambda(x,y)$, we have $E \subseteq E(\lambda) \cup E(\mu)$ for all $\mu \leq 0 \leq \lambda$. If 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 parabola, whereas in the second case E_1 is a canonical 4-point ellipse. By shrinking μ from zero until μ_0 in a similar way we get $E_2 := E(\mu_0)$ which is also a 3-point parabola or a canonical 4-point ellipse. $\square$

Corollary 1. *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* $R \subseteq \mathrm{int}(\cup\mathbf{E})$ *and* $F \cap \mathrm{int}(\cup\mathbf{E}) = \emptyset$ *will be called a* $\mathbf{C}$-cover *for* (R, F)*.*

Corollary 2. *If there is a* $\mathbf{E}^+$*-cover for* (R, F) *of size* k*, then there exists 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. This can easily be made into an $\mathbf{E}^+$-cover for (R, F) of size $O(n)$, so we can conclude that $k_0 \leq \min(m, 2n)$. However, an optimal ellipse cover can be considerably smaller than an optimal circle cover — up to a factor of $\Omega(n)$. Therefore these circles cannot be used as canonical objects.

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.

Lemma 3. $\mathbf{E}_4^+$ *has size* $O(n^2)$ *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 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 3 points of $F(t)$ on their boundary (*3-point disks*), which in turn correspond to F-empty 3-point ellipses y-scaled by $1/t$. Let us consider the dynamic Voronoi diagram, 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. 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 $p = (p_x, p_y) \in F$. Observe that $f_p(x, y, t) = h_p(x, ty, t^2) + x^2 + y^2$ with $h_p(u, v, w) = -2p_x u - 2p_y v + p_y^2 w + p_x^2$, such that the vertices of the lower envelope of the n hyperplanes h_p in $\mathbb{R}^4$ correspond to the vertices of the lower envelope of the original f_p. The lower envelope of n hyperplanes in $\mathbb{R}^4$ has complexity $O(n^2)$ (see, e.g., [13]), so there are indeed only $O(n^2)$ F-empty 4-point ellipses. This lower envelope can also be computed within the same time bounds. □

This bound is tight in the worst case as two sets of points (each of size $n/2$) on the x- and y-axis demonstrate.

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

Proof. Let us argue wlog. that the number of parabolas with a *vertical* axis is $O(n)$. To this end, we map all the points $p = (x, y) \in F$ to $p' = (x, y, x^2)$; this corresponds to lifting F to 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. This shows that there are indeed only $O(n)$ such parabolas and we can compute them all in $O(n \log n)$ time by constructing convex hulls in three dimensions. $\square$

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

Lemma 5. $\mathbf{H}_2$ *has size* $O(n)$ *and can be computed in* $O(n \log n)$ *time.*

4 The covering algorithm

We describe a randomized algorithm that computes a $\mathbf{C}$-cover for (R, F) which consists of $O(k_0 \log k_0)$ canonical objects. The technique was developed in [10, 14, 4, 3]. For clarity of exposition we will assume that $\mathbf{C}$ does not contain parabolas or halfplanes. Below we show how to modify our algorithm to handle these objects as well. The algorithm proceeds in rounds; it works as follows (c is a suitable constant that will be specified in the proof of Lemma 6):

Algorithm 1.
Input: (R, F) and $k > 0$.
Output: A $\mathbf{C}$-cover $\mathbf{E}$ for (R, F) of size $|\mathbf{E}| \leq ck \log k$, if $k \geq k_0$.

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 $V = \{E \in \mathbf{C} \mid q \in E\}$.
5. If $w(V) \leq w(\mathbf{C})/2k$ this round is declared to be *successful* and the weight of all $E \in V$ is doubled.
6. Goto Step 2.

Lemma 6. *If $k \geq k_0$ then*

1. *the probability that a round is successful is at least 1/2 and*
2. *the number of successful rounds is at most $4k_0 \log(n^2/k_0) \leq 8k \log n$.*

Proof. 1. Let $\epsilon := 1/2k$ and consider the range space $\mathcal{S} = (F, \mathbf{C})$. This range space clearly has finite VC-dimension. For the appropriate choice of c, c.f. [8, 11], a random sample $\mathbf{E}$ of size $ck \log k$ from $\mathbf{C}$ is an ϵ-net for $\mathcal{S}$ wrt. the weight function w with probability at least $1/2$. Thus for any $X \subseteq \mathbf{C}$ with $w(X) \geq \epsilon w(\mathbf{C})$ it follows that $\mathbf{E} \cap X \neq \emptyset$. Now if $\mathbf{E}$ is indeed an ϵ-net, we can conclude that $w(V) \leq \epsilon w(\mathbf{C})$, since $\mathbf{E} \cap V = \emptyset$, so the round is successful.

2. In each successful round the total weight $w(\mathbf{C})$ increases by a factor of at most $(1+\epsilon) \leq e^{\epsilon} \leq 2^{3/4k} \leq 2^{3/4k_0}$. Thus, after s successful rounds $w(\mathbf{C}) \leq n^2 2^{3s/4k_0}$. Let $\mathbf{E}_0$ be an optimal $\mathbf{C}$-cover. Since $\mathbf{E}_0$ covers R, clearly $\mathbf{E}_0 \cap V \neq \emptyset$, 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 $\sum_{E \in \mathbf{E}_0} d_E \geq s$, and we can conclude $w(\mathbf{E}_0) = \sum_{E \in \mathbf{E}_0} 2^{d_E} \geq k_0 2^{s/k_0}$, where the last inequality follows with Jensen's inequality. Since $w(\mathbf{E}_0) \leq w(\mathbf{C})$ we finally get $s \leq 4k_0 \log(n^2/k_0)$. □

If $k \geq k_0$ we can view a single round of algorithm 1 as a Bernoulli experiment with success probability at least 1/2. Thus we can apply a suitable Chernoff bound and conclude that the probability that the total number of rounds exceeds $8k \log(n)$ by a factor of t is $O(2^{-t})$.

Now consider the following algorithm (call it Algorithm 2): Given k and $\delta > 0$, we run algorithm 1 for up to $8k \log(n) \log(1/\delta)$ rounds. If we do not find a cover of size at most $ck \log k$ within that number of rounds, we halt. 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 terminates after $8k \log(n) \log(1/\delta)$ rounds, and if $k \geq k_0$ it returns a cover of size at most $ck \log k$ with probability at least $1 - \delta$.*

Since the value of k_0 is not know in general, we have to perform an exponential search for it: To this end, we run algorithm 2 for $k = 1, 2, 4, 8, \ldots$ until it finds a cover (call this procedure Algorithm 3). We get a cover of size at most $2ck_0 \log k_0$ if the algorithm is successful in the $\lceil \log k_0 \rceil$-th step of the exponential search. 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 $O(k_0 \log(n)\ \log(1/\delta))$ rounds a cover of size at most $2ck_0 \log k_0$ with probability at least $1 - \delta$.*

4.1 Implementation

It remains to devise efficient means and data structures to maintain the weights of the objects in $\mathbf{C}$ such that they allow efficient sampling according to w. Moreover we have to specify how to check whether a candidate sample $\mathbf{E}$ constitutes a cover.

Random sampling. 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:

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

where $0 < a < 1$ and $b, c, d \in \mathbb{R}$. The ellipse E contains a point $p = (x, y)$ iff $g(x, y) \leq 0$. 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 efficiently pick an ellipse at random and to maintain the weights efficiently we store $\mathbf{C}'$ in a partition tree data structure: The partition tree of [12] for these $O(n^2)$ points can be constructed in $O(n^2 \log n)$ time, $O(n^2)$ space, and allows halfspace range queries to be answered in time $O(n^{3/2} \log^{O(1)} n)$. The first level of the tree stores a simplicial partition of size $O(n^{3/2})$, where each simplex represents $n^{1/2}$ points. Recursively, a simplex representing r points stores a simplicial partition of size $O(r^{3/2})$. 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 one. 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. To this end we have to double the weights of all points in $\mathbf{C}'$ that lie below h_q. This can be done using the halfspace range query algorithm of [12] 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 below h_q. So the doubling of the weights can be done in $O(n^{3/2} \log^{O(1)} n)$ time.

In order to efficiently pick an ellipse at random 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 each child of v, which corresponds to a simplex in the simplicial partition that v represents, we associate an interval on the real positive line whose length equals the weight of the simplex divided by all factors on the path from v (not including v) to the root, such that all intervals of all children together form a partition of the interval $[0, s_v]$. We store this partition in v as a sorted list. This allows us to go to a random branch in $O(\log n)$ time. During a weight doubling step we can maintain these interval partitions at asymptotically no extra cost since during a query we touch the children of each node that we visit in the recursion anyway. In order to pick an ellipse at random we find a random path from the root to the leaf which requires $O(\log n \log \log n)$ time.

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 $k_1 \leq \sqrt{m}$ 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 $q \in R \setminus \cup \mathbf{E}$ 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 $O(k_0 \log(n) \log(1/\epsilon))$ rounds
   (a) $O(n^{3/2} \log^{O(1)} n)$ time for the weight update and the sampling step, and
   (b) $O((\sqrt{m} + k_0 \log k_0) \sqrt{m} \log m)$ time for the verification step.

Theorem 3. *For any $\epsilon > 0$ algorithm 3 computes with probability at least $1 - \epsilon$ in $O(n^2 \log n + k_0 \log(n) \log(1/\epsilon)(n^{3/2} \log^{O(1)} n + (\sqrt{m} + k_0 \log k_0)\sqrt{m} \log m)) = \tilde{O}(n^2 + k_0 n^{3/2} + k_0 m + k_0^2 \sqrt{m})$ time a cover of size at most $2ck_0 \log k_0$.*

4.2 Handling degenerate cases

To finish the description of our approximation algorithm we need to clarify a few points. First of all 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 equation (1) if we allow $0 \leq a \leq 1$. Therefore the algorithm we just described can handle them without any modifications.

The case of halfplanes is slightly more complicated. However, we can adopt the basic techniques that work for parabolas and ellipses. 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. Thus 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 the 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, the halfplanes and parabolas in a $\mathbf{C}$-cover can be replaced by the smallest enclosing axis-parallel ellipses of the points covered by the corresponding canonical objects that respect F. The total time required by this step is $O((m+n)k_0 \log k_0)$ which is dominated by the runtime of the approximation algorithm.

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 $\cup\mathbf{E} := \cup_{E \in \mathbf{E}} E$ covers R and respects F, i.e., $R \subseteq \cup\mathbf{E}$ and $F \cap \mathrm{int}(\cup\mathbf{E}) = \emptyset$. All our previous arguments and algorithms carry over to this setting. In particular we can compute a non-strict $\mathbf{C}$-cover for (R, F) of size $O(k_0 \log k_0)$ 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. We defer the — quite technical — details of this step for a full version of the paper.

4.3 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. Let therefore a connected pixel pattern R be given. We identify a pixel with its center, and assume that pixels lie on a grid. Let F be the set of grid points not in R that are one pixel away from the boundary of R. Let $n = |F|$, which yields $m = |R| = O(n^2)$. Now we can employ Theorem 3 and obtain a $O(k_0 \log k_0)$ cover in expected $\tilde{O}(n^2 k_0)$ time. Since every connected horizontal or vertical sequence of points of R is always bounded from both sides by a point of F in this setting, we can conclude that halfplanes or parabolas cannot occur in a cover, so we need not take the trouble to handle these special cases.

Acknowledgments. We would like to 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. 11th ACM-SIAM Sympos. 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. H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.*, 14:263–279, 1995.
4. K. L. Clarkson. Algorithms for polytope covering and approximation. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes Comput. Sci.*, pages 246–252. Springer-Verlag, 1993.

5. J. C. Culberson and R. A. Reckhow. Covering polygons is hard. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 601–611, 1988.
6. A. Efrat, F. Hoffmann, K. Kriegel, C. Schultz, and C. Wenk. Geometric algorithms for the analysis of 2d-electrophoresis gels. In *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 114–123, Montreal, Canada, 2001.
7. J. Garrels. The QUEST system for quantitative analysis of 2D gels. *J. Biological Chemistry*, 264:5269–5282, 1989.
8. D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.
9. D. S. Hochbaum. Approximation algorithms of the set covering and vertex cover problems. *SIAM J. Comput.*, 11(3):555–556, 1982.
10. N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 68–77, 1987.
11. J. Matoušek. Cutting hyperplane arrangements. *Discrete Comput. Geom.*, 6:385–406, 1991.
12. J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.*, 8:315–334, 1992.
13. M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, New York, 1995.
14. E. Welzl. Partition trees for triangle counting and other range searching problems. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 23–33, 1988.
15. M. R. Wilkins, K. L. Williams, R. D. Appel, and D. F. Hochstrasser, editors. *Proteome Research: New Frontiers in Functional Genomics*. Springer-Verlag, 1997.