Maximizing the Area of Overlap of two Unions of Disks under Rigid Motion

Mark de Berg

Sergio Cabello

Panos Giannopoulos

Remco C. Veltkamp

institute of information and computing sciences, utrecht university

technical report UU-CS-2004-014

www.cs.uu.nl

Maximizing the Area of Overlap of two Unions of Disks under Rigid Motion

Mark de Berg * Sergio Cabello†, ‡ Panos Giannopoulos§, ¶
Remco C. Veltkamp§

May 4, 2004

Abstract

Let A and B be two sets of n resp. m $(m \geq n)$ disjoint unit disks in the plane. We consider the problem of finding a rigid motion of A that maximizes the total area of its overlap with B. The function describing the area of overlap is quite complex, even for combinatorially equivalent translations, and hence, we turn our attention to approximation algorithms. First, we give a deterministic $(1-\epsilon)$-approximation algorithm for the maximum area of overlap under rigid motion that runs in $O((n^2m^2/\epsilon^3)\log m))$ time. If Δ is the diameter of set A, we get an $(1-\epsilon)$-approximation in $O(\frac{m^2n^{4/3}\Delta^{1/3}\log n \log m}{\epsilon^3})$ time. Under the condition that the maximum is at least a constant fraction of the area of A, we give a probabilistic $(1-\epsilon)$-approximation algorithm that runs in $O((m^2/\epsilon^4)\log(m/\epsilon)\log^2 m)$ time and succeeds with high probability. Our algorithms generalize to the case where A and B consist of possibly intersecting disks of different radii provided that (i) the ratio of the radii of any two disks in $A \cup B$ is bounded, and (ii) within each set, the maximum number of disks with a non-empty intersection is bounded.

Keywords: Geometric Optimization, Approximation Algorithms, Shape Matching, Area of Overlap, Unions of Disks, Rigid Motion

1 Introduction

We study the following *shape matching* problem: given two sets A and B of disks in the plane, we wish to find a rigid motion that maximizes their area of overlap. Our main goal is to match two shapes, each being expressed as a union of disks; thus the overlap we want to maximize is the

*Faculty of Mathematics and Computing Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. Email: m.t.d.berg@TUE.nl.

†IMFM, Department of Mathematics, Jadranska 19, SI-1000 Ljubljana, Slovenia. Email: sergio.cabello@imfm.uni-lj.si

‡Research conducted at IICS, Utrecht University, The Netherlands, and partially supported by the Cornelis Lely Stichting.

§Institute of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands. Email: {panos, Remco.Veltkamp}@cs.uu.nl.

¶Supported by the Dutch Technology Foundation STW, project no. UIF 5055 and a Marie Curie Fellowship of the EC programme 'Combinatorics, Geometry, and Computation', FU Berlin, HPMT-CT-2001-00282.

overlap between the two unions (which is not the same as the sum of overlaps of the individual disks). In the most general setting we assume the following: (i) the largest disk is only a constant times larger than the smallest one, and (ii) any disk in A intersects only a constant number of other disks in A, and the same holds for B.

Motivated by applications in matching shapes that are 'expressed' as unions of convex objects and weighted point set matching, de Berg et al. [dBGK+03] examined the problem of maximizing the area of overlap of two unions of either convex homothets or fat objects under translation. They gave a $(1-\epsilon)$-approximation algorithm that runs in $O((nm/\epsilon^2)\log(m/\epsilon))$ time, where n and m are the sizes of the sets. One of the open problems they mentioned is that of extending the above approximation results to other transformation groups, most notably, rigid motions.

Recently, Cheong et al. [CEHP04] presented a general and elegant technique for solving problems where the goal is to maximize the area of some region that depends on a multi-dimensional parameter. They observed that this technique can be directly applied to our problem, and gave an almost linear, probabilistic approximation algorithm that computes the maximum area of overlap under translations up to an absolute error with high probability. When the maximum overlap is at least a constant fraction of the area of one of the two sets, the absolute error is in fact a relative error. This is usually good enough for shape matching, since if two shapes are quite dissimilar we usually do not care about how bad the match exactly is. A direct application of this technique to rigid motions gives an $O((m^2/\epsilon^6)\log(m/\epsilon)\log^3 m)$ time algorithm that requires the computation of intersection points of algebraic curves of degree six, which is not very practical.

Our contributions are the following. First, in Section 2 ,we show that the maximum number of combinatorially distinct rigid motions of A with respect to B is $O(n^3m^2)$. Moreover, the function describing the area of overlap is quite complex, even for combinatorially equivalent placements. Therefore, we focus on approximation algorithms. Our algorithms are given in the remaining sections. For the sake of clarity we describe the algorithms for the case of disjoint unit disks. It is not hard to adapt them to sets of disks satisfying assumptions (i) and (ii) above; the necessary changes are described in Section 6. For any $\epsilon > 0$, our algorithms can compute a $(1-\epsilon)$-approximation of the optimum overlap. First, we present a deterministic algorithm which runs in $O((n^2m^2/\epsilon^3)\log m)$ time. If Δ is the diameter of set A—recall that we are dealing with unit disks—the running time of the latter becomes $O((m^2n^{4/3}\Delta^{1/3}/\epsilon^3)\log n \log m)$, which yields an improvement when $\Delta = o(n^2/\log^3 n)$. Note that in many applications the union will be connected, which implies that the diameter will be $O(n)$. If the area of overlap is a constant fraction of the area of the union of A, which is the same condition Cheong et al. need, we can get a probabilistic algorithm that runs in $O((m^2/\epsilon^4)\log(m/\epsilon)\log^2 m)$ time, and succeeds with high probability.

Our algorithms for rigid motion are based on a simple two-step framework in which an approximation of the best translation is followed by an approximation of the best rotation. This way, we first achieve an absolute error on the optimum, which we then turn into a relative error using the lower bound theorem by de Berg et al. [dBGK+03, Theorem 5]; for completeness This theorem, which we reproduce in Section 2 for completeness, gives a lower bound on the maximum area of overlap under translations, expressed in the number of pairs of disks that contribute to that area. The deterministic algorithm employs a clever sampling of transformation space, directed by some special properties of the function of the area of overlap of two disks. The probabilistic algorithm is a combination of sampling of translation space using a uniform grid, random sampling of both input sets, and the technique by Cheong et al.

2 Basic properties of the overlap function

We start by introducing some notation. Let $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_m\}$, be two sets of disjoint unit disks in the plane, with $n \leq m$. We consider the disks to be closed. Both A and B lie in the same two-dimensional coordinate space, which we call the *work space*; their initial position is denoted simply by A and B. We consider B to be fixed, while A can be translated and/or rotated relative to B.

Let $\mathcal{I}$ be the infinite set of all possible rigid motions—also called isometries—in the plane; we call $\mathcal{I}$ the *configuration space*. We denote by R_θ a rotation about the origin by some angle $\theta \in [0, 2\pi)$ and by $T_{\vec{t}}$ a translation by some $\vec{t} \in \mathbb{R}^2$. It will be convenient to model the space $[0, 2\pi)$ of rotations by points on the circle S^1. For simplicity, rotated only versions of A are denoted by $A(\theta) = \{A_1(\theta), \ldots, A_n(\theta)\}$. Similarly, translated only versions of A are denoted by $A(\vec{t}) = \{A_1(\vec{t}), \ldots, A_n(\vec{t})\}$. Any rigid motion $I \in \mathcal{I}$ can be uniquely defined as a translation followed by a rotation, that is, $I = I_{\vec{t},\theta} = R_\theta \circ T_{\vec{t}}$, for some $\theta \in S^1$ and $\vec{t} \in \mathbb{R}^2$. Alternatively, a rigid motion can be seen as a rotation followed by some translation; it will be always clear from the context which definition is used. In general, transformed versions of A are denoted by $A(\vec{t}, \theta) = \{A_1(\vec{t}, \theta), \ldots, A_n(\vec{t}, \theta)\}$ for some $I_{\vec{t},\theta} \in \mathcal{I}$.

Let $\mathrm{Int}(C), V(C)$ be, respectively, the interior and area of a compact set $C \in \mathbb{R}^2$, and let $\mathcal{V}_{ij}(\vec{t}, \theta) = V(A_i(\vec{t}, \theta) \cap B_j)$. The *area of overlap* of $A(\vec{t}, \theta)$ and B, as $\vec{t}, \theta$ vary, is a function $\mathcal{V} : \mathcal{I} \to \mathbb{R}$ with $\mathcal{V}(\vec{t}, \theta) = V((\bigcup A(\vec{t}, \theta)) \cap (\bigcup B))$. Thus the problem that we are studying can be stated as follows:

Given two sets A, B, defined as above, compute a rigid motion $I_{\vec{t}_{opt},\theta_{opt}}$ that maximizes $\mathcal{V}(\vec{t}, \theta)$.

Let $d_{ij}(\vec{t}, \theta)$ be the Euclidean distance between the centers of $A_i(\vec{t}, \theta)$ and B_j. For simplicity, we write $\mathcal{V}(\vec{t}), \mathcal{V}_{ij}(\vec{t}), d_{ij}(\vec{t})$ when θ is fixed and $\mathcal{V}(\theta), \mathcal{V}_{ij}(\theta), d_{ij}(\theta)$ when $\vec{t}$ is fixed. Also, let r_i be the Euclidean distance of A_i's center to the origin. The Minkowski sum of two planar sets A and B, denoted by $A \oplus B$, is the set $\{p_1 + p_2 : p_1 \in A, p_2 \in B\}$. Similarly the Minkowski difference $A \ominus B$ is the set $\{p_1 - p_2 : p_1 \in A, p_2 \in B\}$.

Theorem 1 *Let A be a set of n disjoint unit disks in the plane, and B a set of m disjoint unit disks, with $n \leq m$. The maximum number of combinatorially distinct rigid motions of A with respect to B is $O(n^3 m^2)$.*

Proof: Let us assume for a moment that A is first rotated about the origin by some fixed angle $\theta \in [0, 2\pi)$. We define $T_{ij}(\theta) = B_j \ominus A_i(\theta)$; $\mathcal{V}_{ij}(\vec{t}, \theta) > 0$ if and only if $\vec{t} \in \mathrm{Int}(T_{ij}(\theta))$. Let $T(A, B)(\theta) = \{T_{ij}(\theta) : A_i \in A \text{ and } B_j \in B\}$. Then, $\mathcal{V}(\vec{t}, \theta) > 0$ if and only if $\vec{t} \in \mathrm{Int}(T(A, B)(\theta))$. The boundaries of the Minkowski differences $T_{ij}(\theta) \in T(A, B)(\theta)$ induce a planar subdivision $\mathcal{T}(\theta)$. Each cell in this arrangement is a set of combinatorially equivalent translations of $A(\theta)$ relative to B, that is, the set of all overlapping pairs $(A_i(\vec{t}, \theta), B_j)$ is the same for all $\vec{t}$ in the cell. $\mathcal{T}(\theta)$ can be non-simple and non-connected, and its maximum complexity is $\Theta(n^2 m)$ [dBGK+03].

As θ varies, the combinatorial structure of $\mathcal{T}(\theta)$ changes: each $T_{ij}(\theta)$ rotates about the center of B_j and, as a result, new cells are created or existing cells dissappear. Such a change occurs in one of the following two cases: (i) when two arcs in $\mathcal{T}(\theta)$ become tangent at some θ (double event) or (ii) three arcs in $\mathcal{T}(\theta)$ intersect at a point (triple event). By the analysis of Chew et al. [CGH+97], the number of double events is $O(n^2 m^2)$ and the number of triple events is $O(n^3 m^2)$. Thus, the complexity of the configuration space[1] is $O(n^3 m^2)$ as well. □

[1] Abusing the terminology slightly, we will sometimes use the term 'configuration space' when we are actually referring to the decomposition of the configuration space induced by the infinite family of sets $T(A, B)(\theta)$ for all $\theta \in [0, 2\pi)$.

This theorem implies that explicitly computing the subdivision of the configuration space into cells with combinatorially equivalent placements is highly expensive. Moreover, the computation for rigid motions can cause non-trivial numerical problems since it requires the computation of intersection points between algebraic curves of degree six and circles [AG99]. Finally, the optimization problem in a cell of this decomposition is far from easy: one has to maximize a function consisting of a linear number of terms; see de Berg et al. for details. Therefore we turn our attention to approximation algorithms. The following theorem, which gives a lower bound on the maximum area of overlap, will be instrumental in obtaining a relative error; see de Berg et al. for a proof.

Theorem 2 *Let $A = \{A_1, \dots, A_n\}$ and $B = \{B_1, \dots, B_m\}$ be two sets of disjoint unit disks in the plane. Let $\vec{t}_{opt}$ be the translation that maximizes the area of overlap $\mathcal{V}(\vec{t})$ of $A(\vec{t})$ and B over all possible translations $\vec{t}$ of set A. If k_{opt} is the number of overlapping pairs $A_i(\vec{t}_{opt}), B_j$, then $\mathcal{V}(\vec{t}_{opt})$ is $\Theta(k_{opt})$.*

3 The rotational case

This section considers the following restricted scenario: set B is fixed, and set A can be rotated around the origin. This will be used in the next section, where we consider general rigid motions.

Observe that this problem has a one-dimensional configuration space: the angle of rotation. Consider the function $\mathcal{V} : [0, 2\pi) \to \mathbb{R}$ with

$$\mathcal{V}(\theta) := V((\bigcup A(\theta)) \cap (\bigcup B)) = \sum_{A_i \in A, B_j \in B} \mathcal{V}_{ij}(\theta).$$

For now, our objective is to guarantee an absolute error on $\mathcal{V}$ rather than a relative one. We start with a result that bounds the difference in overlap for two relatively similar rotations. Recall that r_i is the distance of A_i's center to the origin.

Lemma 3 *Let A_i, B_j be any fixed pair of disks. For any given $\delta > 0$ and any θ_1, θ_2 for which $|\theta_1 - \theta_2| \leq \delta/(2r_i)$, we have $|\mathcal{V}_{ij}(\theta_1) - \mathcal{V}_{ij}(\theta_2)| \leq 2\delta$.*

Proof: Without loss of generality, we assume that $\theta_1 = 0$ and that A_i is centered at $(r_i, 0)$ with $r_i > 0$; see Figure 1. We want to see that $V(A_i \cap B_j) - V(A_i(\theta) \cap B_j) \leq 2\delta$ for any $0 \leq \theta \leq \delta/(2r_i)$. Consider the function $v(\theta) = V(A_i \cap A_i(\theta))$ with $\theta \in [0, \pi/2]$. We will prove that if $0 \leq \theta \leq \delta/(2r_i)$ then $v(\theta) \geq \pi - \delta$, and therefore $V(A_i \setminus A_i(\theta)) = V(A_i(\theta) \setminus A_i) \leq \delta$. Using that for any sets X, Y we have $V(X) - V(Y) = V(X \setminus Y) - V(Y \setminus X)$, then for any $0 \leq \theta \leq \delta/(2r_i)$ it holds

$$\begin{aligned}
&|V(A_i \cap B_j) - V(A_i(\theta) \cap B_j)| \\
&\qquad = |V((A_i \cap B_j) \setminus (A_i(\theta) \cap B_j)) - V((A_i(\theta) \cap B_j) \setminus (A_i \cap B_j))| \\
&\qquad \leq |V((A_i \cap B_j) \setminus (A_i(\theta) \cap B_j))| + |V((A_i(\theta) \cap B_j) \setminus (A_i \cap B_j))| \\
&\qquad \leq V(A_i \setminus A_i(\theta)) + V(A_i(\theta) \setminus A_i) \\
&\qquad \leq 2\delta,
\end{aligned}$$

and the lemma follows.

We will show that $v(\theta) \geq \pi - \delta$ using the mean-value theorem. The center of $A_i(\theta)$ is positioned at $(r_i \cos(\theta), r_i \sin(\theta))$ and the distance between the centers of A_i and $A_i(\theta)$ is

$$\sqrt{r_i^2(1 - \cos\theta)^2 + r_i^2 \sin^2\theta} = r_i\sqrt{2(1 - \cos(\theta))}.$$

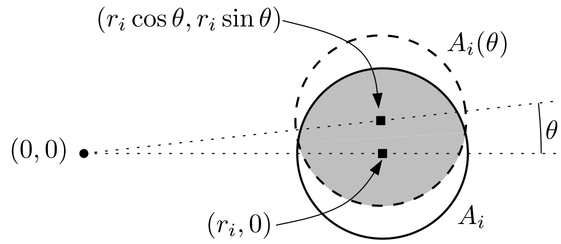

Figure 1: Notation in Lemma 3. The area of the grey region corresponds to $v(\theta)$.

The area of overlap of two unit disks whose centers are d apart is

$$2\arccos\frac{d}{2} - \frac{d\sqrt{4-d^2}}{2},$$

and therefore we get

$$\begin{aligned}
\frac{\partial v(\theta)}{\partial \theta} &= \frac{\partial v(\theta)}{\partial d} \cdot \frac{\partial d}{\partial \theta} \\
&= -\sqrt{4 - 2r_i^2(1-\cos(\theta))} \cdot \frac{r_i \sin(\theta)}{\sqrt{2(1-\cos(\theta))}} \\
&= -r_i\sqrt{2 + 2\cos(\theta) - r_i^2 \sin^2(\theta)} \geq -2r_i,
\end{aligned}$$

where in the last inequality we used $2 \geq 2\cos(\theta) - r_i \sin^2(\theta)$. We conclude that if $0 \leq \theta \leq \delta/(2r_i)$ then $\partial v(\theta)/\partial\theta \geq -\delta/\theta$.

Using the mean-value theorem we see that, for any $\theta \in [0, \delta/(2r_i)]$ there exists $\theta' \in [0,\theta]$ such that

$$\frac{v(\theta) - v(0)}{\theta - 0} = \frac{v(\theta) - \pi}{\theta} = \frac{\partial v(\theta')}{\partial \theta}.$$

Since, $0 \leq \theta' \leq \delta/(2r_i)$, we have $\partial v(\theta')/\partial\theta \geq -\frac{\delta}{\theta}$ and so we conclude that $v(\theta) - \pi \geq -\delta$. □

For a pair A_i, B_j, we define the interval $R_{ij} = \{\theta \in [0, 2\pi) : A_i(\theta) \cap B_j \neq \emptyset\}$ on S^1, the circle of rotations. We denote the length of R_{ij} by $|R_{ij}|$. Instead of computing $\mathcal{V}_{ij}(\theta)$ at each $\theta \in R_{ij}$, we would like to sample it at regular intervals whose length is at most $\delta/(2r_i)$. At first, it looks as if we would have to take an infinite number of sample points as $r_i \to \infty$. However, as the following lemma shows, $|R_{ij}|$ decreases as r_i increases, and the number of samples we need to consider is bounded.

Lemma 4 *For any A_i, B_j with $r_i > 0$, and any given given $\delta > 0$, we have $|R_{ij}|/(\delta/(2r_i)) = O(1/\delta)$.*

Proof: Without loss of generality, we can assume that A_i is centered at $(r_i, 0)$ and B_j is centered at $(r_j, 0)$. Note that the distance between the center of $A_i(\theta)$ and B_j is

$$d_{ij}(\theta) = \sqrt{(r_i\cos\theta - r_j)^2 + (r_i \sin\theta)^2} = \sqrt{r_i^2 + r_j^2 - 2r_i r_j \cos\theta}.$$

Under these assumptions, R_{ij} is of the form $[-\theta_{ij}, \theta_{ij}]$, where θ_{ij} is the largest value for which $A_i(\theta_{ij}) \cap B_j \neq \emptyset$, that is, $d_{ij}(\theta_{ij}) = 2$. We have $\theta_{ij} = \arccos\frac{r_i^2 + r_j^2 - 4}{2r_i r_j}$.

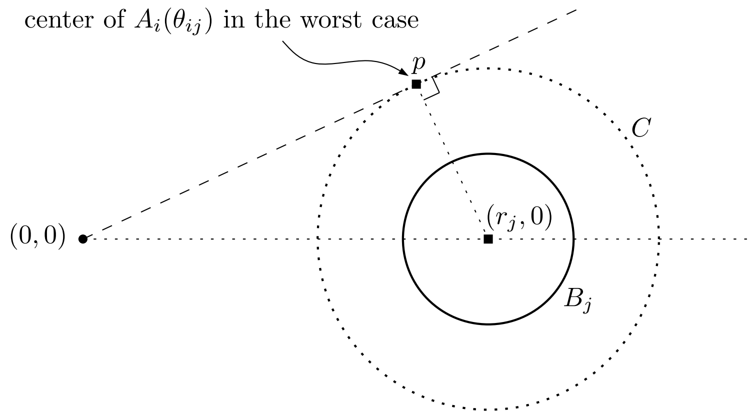

Figure 2: Notation in Lemma 4. The center of $A_i(\theta_{ij})$ is placed in the circle C. Therefore, θ_{ij} is maximized for the dashed line through the origin and tangent to C.

As shown in Figure 2, the center of $A_i(\theta_{ij})$ is always placed on C, the circle of radius two and concentric with B_j. Therefore, the value θ_{ij} is maximized when it equals the slope of the line through the origin and tangent to C. Let p be the point of tangency. Since the triangle $p, (0,0), (r_j, 0)$ is right on p, we conclude that θ_{ij} is maximized when $r_j = \sqrt{r_i^2 + 4}$. Therefore

$$|R_{ij}| = 2 \arccos \frac{r_i^2 + r_j^2 - 4}{2r_i r_j} \leq 2 \arccos \sqrt{1 - \frac{4}{r_i^2 + 4}}.$$

Using L'Hôpital's rule we can compute that

$$\lim_{r_i \to \infty} \frac{|R_{ij}|}{1/r_i} = \lim_{r_i \to \infty} \frac{4}{1 + \frac{4}{r_i^2}} = 4.$$

It follows that the function $|R_{ij}| \cdot r_i$ is bounded for any $r_i > 0$, and so $\frac{|R_{ij}|}{\delta/(2r_i)} = O(1/\delta)$. □

This lemma implies that we have to consider only $O(1/\delta)$ sample rotations per pair of disks. Thus we need to check $O(nm/\delta)$ rotations in total. It seems that we would have to compute all overlaps at every rotation from scratch, but here Lemma 3 comes to the rescue: in between two consecutive rotations θ, θ' defined for a given pair A_i, B_j there may be many other rotations, but if we conservatively estimate the overlap of A_i, B_j as the minimum overlap of θ and θ', we do not loose too much. In Figure 3, algorithm ROTATION is described in more detail; the value $\tilde{\mathcal{V}}(\theta)$ is the conservative estimate of $\mathcal{V}(\theta)$, as just explained.

Lemma 5 *Let θ_{opt} be a rotation that maximizes $\mathcal{V}(\theta)$ and let k_{opt} be the number of overlapping pairs $A_i(\theta_{opt}), B_j$. For any given $\delta > 0$, the rotation θ_{apx} reported by* ROTATION*(A, B, δ) satisfies $\mathcal{V}(\theta_{opt}) - \mathcal{V}(\theta_{apx}) = O(k_{opt}\delta)$, and can be computed in $O((mn/\delta)\log m)$ time.*

Proof: First, we show that $\mathcal{V}(\theta) \geq \tilde{\mathcal{V}}(\theta) \geq \mathcal{V}(\theta) - 2k_\theta\delta$ for any $\theta \in \Theta$ where k_θ is the number of overlapping pairs between $A(\theta)$ and B. That is, $\tilde{\mathcal{V}}$ is a fair approximation of $\mathcal{V}$ from below for the values in Θ.

By checking whether $\mathcal{V}_{ij}$ increases or decreases at θ_{ij}^s and adding the appropriate value to $\tilde{\mathcal{V}}(\theta)$, each pair A_i, B_j contributes $\mathcal{V}_{ij}(\theta_{ij}^s) \leq \mathcal{V}_{ij}(\theta)$ to $\tilde{\mathcal{V}}(\theta)$ for some θ_{ij}^s for which $|\theta - \theta_{ij}^s| \leq \delta/(2r_i)$. By Lemma 3 we have $\mathcal{V}_{ij}(\theta) - \mathcal{V}_{ij}(\theta_{ij}^s) \leq 2\delta$. Thus, in total, $0 \leq \mathcal{V}(\theta) - \tilde{\mathcal{V}}(\theta) \leq 2k_\theta\delta$.

ROTATION(A, B, δ):

1. For each pair of disks $A_i \in A$ and $B_j \in B$, choose a set $\Theta_{ij} := \{\theta_{ij}^1, \dots, \theta_{ij}^{s_{ij}}\}$ of rotations as follows. First put the midpoint of R_{ij} in Θ_{ij}, and then put all rotations in Θ_{ij} that are in R_{ij} and are at distance $k \cdot \delta/(2r_i)$ from the midpoint for some integer k. Finally, put both endpoints of R_{ij} in Θ_{ij}. In other words, Θ_{ij} consists of rotations with a uniform spacing of $\delta/(2r_i)$—except for the cases of endpoints whose distance to their neighbor rotations is less than $\delta/(2r_i)$—with the midpoint of R_{ij} being one of them.
2. Sort the values $\Theta := \bigcup_{i,j} \Theta_{ij}$, keeping repetitions and solving ties arbitrarily. Let $\theta_0, \theta_1, \dots$ be the ordering of Θ. In steps 3 and 4, we will compute a value $\tilde{\mathcal{V}}(\theta)$ for each $\theta \in \Theta$.
3. (a) Initialize $\tilde{\mathcal{V}}(\theta_0) := 0$.
   (b) For each pair $A_i \in A, B_j \in B$ for which $\theta_0 \in R_{ij}$ do:
      - If $\mathcal{V}_{ij}$ is decreasing at θ_0, or θ_0 is the midpoint of R_{ij}, then $\tilde{\mathcal{V}}(\theta_0) := \tilde{\mathcal{V}}(\theta_0) + \mathcal{V}_{ij}(\tilde{\theta}_{ij})$, where $\tilde{\theta}_{ij}$ is the closest value to θ_0 in Θ_{ij} with $\tilde{\theta}_{ij} > \theta_0$.
      - If $\mathcal{V}_{ij}$ is increasing at θ_0, then $\tilde{\mathcal{V}}(\theta_0) := \tilde{\mathcal{V}}(\theta_0) + \mathcal{V}_{ij}(\tilde{\theta}_{ij})$, where $\tilde{\theta}_{ij}$ is the closest value to θ_0 in Θ_{ij} with $\tilde{\theta}_{ij} < \theta_0$.
4. For each θ_l in increasing order of l, compute $\tilde{\mathcal{V}}(\theta_l)$ from $\tilde{\mathcal{V}}(\theta_{l-1})$ by updating the contribution of the pair A_i, B_j defining θ_l, as follows. Let θ_l be the s-th point in Θ_{ij}, that is, $\theta_l = \theta_{ij}^s$
   - If $\mathcal{V}_{ij}$ is increasing at θ_{ij}^s, then $\tilde{\mathcal{V}}(\theta_l) := \tilde{\mathcal{V}}(\theta_{l-1}) - \mathcal{V}_{ij}(\theta_{ij}^{s-1}) + V_{ij}(\theta_{ij}^s)$
   - If $\mathcal{V}_{ij}$ is the midpoint of R_{ij}, then $\tilde{\mathcal{V}}(\theta_l) := \tilde{\mathcal{V}}(\theta_{l-1}) - \mathcal{V}_{ij}(\theta_{ij}^{s-1}) + \mathcal{V}_{ij}(\theta_{ij}^{s+1})$
   - If $\mathcal{V}_{ij}$ is decreasing at θ_{ij}^s, then $\tilde{\mathcal{V}}(\theta_l) := \tilde{\mathcal{V}}(\theta_{l-1}) - \mathcal{V}_{ij}(\theta_{ij}^s) + \mathcal{V}_{ij}(\theta_{ij}^{s+1})$
5. Report the $\theta_{apx} \in \Theta$ that maximizes $\tilde{\mathcal{V}}(\theta)$.

Figure 3: Algorithm ROTATION(A, B, δ).

In a similar fashion, consider now the k_{opt} overlapping pairs of disks at θ_{opt}, and let A_M be the disk furthest from the origin that participates in the optimal solution, i.e. $A_M(\theta_{opt}) \cap (\bigcup B) \neq \emptyset$. Let $\tilde{\theta} \in \Theta$ be the closest value to θ_{opt}. We have

$$|\tilde{\theta} - \theta_{\mathrm{opt}}| \leq \delta/(2r_M) \leq \delta/(2r_i)$$

for all A_i in the optimal solution. Again, according to Lemma 3, the loss per pair A_i, B_j is $\mathcal{V}_{ij}(\theta_{\mathrm{opt}}) - \mathcal{V}_{ij}(\tilde{\theta}) \leq 2\delta$. In total, $\mathcal{V}(\theta_{opt}) - \mathcal{V}(\tilde{\theta}) \leq 2k_{opt}\delta$.

Observe that since both endpoints of every interval R_{ij} are in Θ, no new pairs with non-zero overlap are formed when 'moving' from θ_{opt} to $\tilde{\theta}$. Hence, for our purpose, we can assume that $k_{\tilde{\theta}} = k_{opt}$.

Putting it all together we get

$$\begin{aligned} \mathcal{V}(\theta_{\mathrm{opt}}) - \mathcal{V}(\theta_{apx}) &= \big(\mathcal{V}(\theta_{opt}) - \mathcal{V}(\tilde{\theta})\big) + \big(\mathcal{V}(\tilde{\theta}) - \tilde{\mathcal{V}}(\tilde{\theta})\big) + \big(\tilde{\mathcal{V}}(\tilde{\theta}) - \tilde{\mathcal{V}}(\theta_{apx})\big) \\ &\quad + \big(\tilde{\mathcal{V}}(\theta_{apx}) - V(\theta_{apx})\big) \\ &\leq 2k_{opt}\delta + 2k_{\tilde{\theta}}\delta + 0 + 0 \leq 4k_{opt}\delta. \end{aligned}$$

The running time is dominated by the time to sort the values in Θ. The set Θ consists of $O(nm)$ subsets Θ_{ij}, which each have $O(1/\delta)$ rotations by Lemma 4. Each subset Θ_{ij} can easily be generated as a sorted sequence, so what remains is to merge the sorted sequences, which can be done in $O((nm/\delta)\log m)$ time. $\square$

4 A $(1-\epsilon)$-approximation algorithm for rigid motions

As noted in the introduction, any rigid motion can be described as a translation plus a a rotation about the origin. This is used in the algorithm RigidMotion given in Figure 4. First, we start with the following lemma which implies that, in terms of absolute error, it is not too bad if we choose a translation which is close to the optimal one.

Lemma 6 *Let k be the number of overlapping pairs $A_i(\vec{t},\theta), B_j$ for some $\vec{t} \in \mathbb{R}^2, \theta \in [0, 2\pi)$. For any given $\delta > 0$ and any $\vec{t}' \in \mathbb{R}^2$ for which $|\vec{t} - \vec{t}'| = O(\delta)$, we have $\mathcal{V}(\vec{t}', \theta) = \mathcal{V}(\vec{t}, \theta) - O(k\delta)$.*

Proof: Consider a pair of disks $A_i(\vec{t}, \theta)$ and B_j for which $\mathcal{V}_{ij}(\vec{t}, \theta) \neq 0$. If A_i is translated by $\vec{t}'$, instead of $\vec{t}$, then $d_{ij}(\vec{t}', \theta) - d_{ij}(\vec{t}, \theta) = |\vec{t} - \vec{t}'|$. Observe that the biggest *loss per pair*, $\mathcal{V}_{ij}(\vec{t}, \theta) - \mathcal{V}_{ij}(\vec{t}', \theta)$, occurs when A_i moves in the direction of the line connecting the centers of A_i and B_j, and away from B_j. Since the diameter of both disks is equal to 2, we have that $\mathcal{V}_{ij}(\vec{t}, \theta) - \mathcal{V}_{ij}(\vec{t}', \theta) < 2|\vec{t} - \vec{t}'| = O(\delta)$. We have k such pairs, hence[2] , $\mathcal{V}(\vec{t}, \theta) - \mathcal{V}(\vec{t}', \theta) = O(k\delta)$. □

RigidMotion(A, B, ϵ):

1. Let G be a uniform grid of spacing $c\epsilon$, where c is a suitable constant. For each pair of disks $A_i \in A$ and $B_j \in B$ do:
   (a) Set the center of rotation, i.e. the origin, to be B_j's center by translating B appropriately.
   (b) Let $T_{ij} = B_j \ominus A_i$, and determine all grid points $\vec{t}_g$ of G such that $\vec{t}_g \in T_{ij}$. For each such $\vec{t}_g$ do:
      - run Rotation$(A(\vec{t}_g), B, c'\epsilon)$, where c' is an appropriate constant. Let θ^g_{apx} be the rotation returned. Compute $\mathcal{V}(\vec{t}_g, \theta^g_{apx})$.
2. Report the pair $(\vec{t}_{apx}, \theta_{apx})$ that maximizes $\mathcal{V}(\vec{t}_g, \theta^g_{apx})$.

Figure 4: Algorithm RigidMotion(A, B, ϵ).

Theorem 7 *Let $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_m\}$, with $n \leq m$, be two sets of disjoint unit disks in the plane. Let $I_{\vec{t}_{opt}, \theta_{opt}}$ be a rigid motion that maximizes $\mathcal{V}(\vec{t}, \theta)$. Then, for any given $\epsilon > 0$, RigidMotion(A, B, ϵ) computes a rigid motion $I_{\vec{t}_{apx}, \theta_{apx}}$ such that $\mathcal{V}(\vec{t}_{apx}, \theta_{apx}) \geq (1-\epsilon)\mathcal{V}(\vec{t}_{opt}, \theta_{opt})$ in $O((n^2m^2/\epsilon^3)\log m)$ time.*

Proof: We will show that $\mathcal{V}(\vec{t}_{apx}, \theta_{apx})$ approximates $\mathcal{V}(\vec{t}_{opt}, \theta_{opt})$ up to an absolute error. To convert the absolute error into a relative error, and hence show the algorithm's correctness, we use again Theorem 2.

Let A_{opt} be the set of disks in A that participate in the optimal solution and let $|A_{opt}| = \bar{k}_{opt}$. Since the 'kissing' number of unit open disks is six, we have that $k_{opt} < 6\bar{k}_{opt}$, where k_{opt} is the number of overlapping pairs in the optimal solution. Next, imagine that RigidMotion(A_{opt}, B, ϵ) is run instead of RigidMotion(A, B, ϵ). Of course, an optimal rigid motion for A_{opt} is an optimal rigid motion for A and the error we make by applying a non-optimal rigid motion to A_{opt} bounds the error we make when applying the same rigid motion to A.

Consider a disk $A_i \in A_{opt}$ and an intersecting pair $A_i(\vec{t}_{opt}, \theta_{opt})$, B_j. Since, at some stage, the algorithm will use B_j's center as the center of rotation, and $I_{\vec{t}_{opt}, \theta_{opt}} = R_{\theta_{opt}} \circ T_{\vec{t}_{opt}}$, we have that

[2] Note that by translating A by $\vec{t}'$ instead of $\vec{t}$ and then rotating it by θ, new pairs might appear but this can only decrease the total loss.

$A_i(\vec{t}_{\rm opt}) \cap B_j \neq \emptyset$ if and only if $A_i(\vec{t}_{opt}, \theta_{opt}) \cap B_j \neq \emptyset$. Hence, we have that $\vec{t}_{opt} \in T_{ij}$ and the algorithm will consider some grid translation $\vec{t}_g \in T_{ij} = B_j \ominus A_i$, for which $|\vec{t}_{opt} - \vec{t}_g| = O(\epsilon)$. By Lemma 6 we have $\mathcal{V}(\vec{t}_{opt}, \theta_{opt}) - \mathcal{V}(\vec{t}_g, \theta_{opt}) = O(k_{opt}\epsilon) = O(\bar{k}_{opt}\epsilon)$.

Let $\theta^g_{\rm opt}$ be the optimal rotation for $\vec{t}_g$. Then, $\mathcal{V}(\vec{t}_g, \theta_{\rm opt}) \leq \mathcal{V}(\vec{t}_g, \theta^g_{\rm opt})$. The algorithm computes, in its second loop, a rotation θ^g_{apx} for which $\mathcal{V}(\vec{t}_g, \theta^g_{\rm opt}) - \mathcal{V}(\vec{t}_g, \theta^g_{apx}) = O(k^g_{opt}\epsilon)$, where k^g_{opt} is the number of pairs at the optimal rotation $\theta^g_{\rm opt}$ of $A_{opt}(\vec{t}_g)$. Since we are only considering A_{opt} we have that $k^g_{opt} < 6\bar{k}_{opt}$, thus, $\mathcal{V}(\vec{t}_g, \theta^g_{\rm opt}) - \mathcal{V}(\vec{t}_g, \theta^g_{apx}) = O(\bar{k}_{opt}\epsilon)$.

Now, using the fact that $\mathcal{V}(\vec{t}_g, \theta^g_{apx}) \leq \mathcal{V}(\vec{t}_{apx}, \theta_{apx})$ and that $\bar{k}_{opt} \leq k_{opt}$, and putting it all together we get

$$\begin{aligned}
\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt}) - \mathcal{V}(\vec{t}_{apx}, \theta_{apx}) &= \big(\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt}) - \mathcal{V}(\vec{t}_g, \theta_{\rm opt})\big) + \big(\mathcal{V}(\vec{t}_g, \theta_{\rm opt}) - \mathcal{V}(\vec{t}_g, \theta^g_{\rm opt})\big) \\
&\quad + \big(\mathcal{V}(\vec{t}_g, \theta^g_{\rm opt}) - \mathcal{V}(\vec{t}_g, \theta^g_{apx})\big) + \big(\mathcal{V}(\vec{t}_g, \theta^g_{apx}) - \mathcal{V}(\vec{t}_{apx}, \theta_{apx})\big) \\
&< O(\bar{k}_{opt}\epsilon) + 0 + O(\bar{k}_{opt}\epsilon) + 0 = O(k_{opt}\epsilon).
\end{aligned}$$

Since the optimal rigid motion can be also defined as a rotation followed by some translation, Theorem 2 holds for $\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt})$ as well. Thus, $\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt}) = \Theta(k_{opt})$ and the approximation bound follows.

Finally, the running time of the algorithm is dominated by its first step. We can compute $\mathcal{V}(\vec{t}_g, \theta^g_{apx})$ by a simple plane sweep in $O(m \log m)$ time. Since there are $\Theta(\epsilon^{-2})$ grid point in each T_{ij}, each execution of the loop in the first step takes $O(m + 1/\epsilon^2 + (1/\epsilon^2)(nm/\epsilon)\log m + (1/\epsilon^2)m \log m) = O((nm/\epsilon^3)\log m)$ time. The step is executed nm times, thus the algorithm runs in $O((n^2m^2/\epsilon^3)\log m)$ time. ◻

4.1 An improvement for sets with small diameter

We can modify the algorithm such that its running time depends on the diameter Δ of the set A. The main idea is to convert our algorithm into one that is sensitive to the number of pairs of disks in A and B that have approximately the same distance, and then use the combinatorial bounds by Gavrilov et al. [GIMV03]. Namely, we will use the following result (note that an extra $\log n$ factor is missing in the reference due to a typographic error).

Lemma 8 *[GIMV03, Theorem 4.1] Given a S set of n points whose closest pair is at distance at least 2, there are $O(n^{4/3}t^{1/3}\log n)$ pairs of points in S whose distance is in the range $[t-4, t+4]$.*

This lemma and a careful implementation of ROTATION allows us to improve the analysis of the running time of RIGIDMOTION for small values of Δ. In many applications it is reasonable to assume bounds of the type $\Delta = O(n)$ [GIMV03], and therefore the result below is relevant. For example, if $\Delta = O(n)$ this result shows that we can compute a $(1-\epsilon)$-approximation in $O((m^2n^{5/3})/\epsilon^3 \log n \log m)$ time.

Theorem 9 *Let $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_m\}, n \leq m$ be two sets of disjoint unit disks in the plane. Let Δ be the diameter of A, and let $I_{\vec{t}_{opt}, \theta_{opt}}$ be the rigid motion maximizing $\mathcal{V}(\vec{t}, \theta)$. For any $\epsilon > 0$, we can find in $O((m^2n^{4/3}\Delta^{1/3}/\epsilon^3)\log n \log m)$ time a rigid motion $I_{\vec{t}_{apx}, \theta_{apx}}$ such that $\mathcal{V}(\vec{t}_{apx}, \theta_{apx}) \geq (1-\epsilon)\mathcal{V}(\vec{t}_{opt}, \theta_{opt})$.*

Proof: Observe that when RIGIDMOTION calls ROTATION the origin is set at the center of some B_j, and some A_i intersects B_j. We denote by c_{A_i} the center of A_i and similarly by c_{B_j} the center of B_j. Inside ROTATION, the pairs of disks $A_{i'}$ and $B_{j'}$ such that $R_{i'j'} \neq \emptyset$ must satisfy

$$d(c_{B_j}, c_{B_{j'}}) - 4 \leq d(c_{A_i}, c_{A_{i'}}) \leq d(c_{B_j}, c_{B_{j'}}) + 4$$

We store in a balanced tree the distances from the origin to the centers of A. This can be done in $O(n \log n)$ time, and then we can report, for any given pair $B_j, B_{j'} \in B$ and any $A_i \in A$, all the pairs $A_i, A_{i'} \in A$ satisfying the relation above in $O(k_{j,j',i} + \log n)$ time, where $k_{j,j',i}$ is the number of reported pairs. Therefore, when we have fixed $\vec{t_g} \in T_{ij}$ in RIGIDMOTION, we can implement the call to ROTATION$(A(\vec{t_g}), B, c\epsilon)$ in $O(m \log n + \frac{\sum_{j'} k_{j,j',i}}{\epsilon} \log m)$ time.

If A has diameter Δ, then Lemma 8 implies $\sum_i k_{j,j',i} = O(n^{4/3}\Delta^{1/3} \log n)$. This means that, overall, RIGIDMOTION can be implemented in

$$O\left(\frac{1}{\epsilon^2} \sum_{i,j} (m \log n + \frac{\sum_{j'} k_{j,j',i}}{\epsilon} \log m)\right) =$$

$$O\left(\frac{nm^2 \log n}{\epsilon^2} + \frac{\sum_{i,j,j'} k_{j,j',i}}{\epsilon^3} \log m\right) =$$

$$O\left(\frac{nm^2 \log n}{\epsilon^2} + \frac{\sum_{j,j'} \sum_i k_{j,j',i}}{\epsilon^3} \log m\right) =$$

$$O\left(\frac{nm^2 \log n}{\epsilon^2} + \frac{m^2 n^{4/3} \Delta^{1/3} \log n}{\epsilon^3} \log m\right) = O\left(\frac{m^2 n^{4/3} \Delta^{1/3} \log n \log m}{\epsilon^3}\right) \text{ time.}$$

□

5 A Monte Carlo algorithm

In this section we present a Monte Carlo algorithm that computes a $(1-\epsilon)$-approximation for rigid motions in $O((m^2/\epsilon^4)\log(m/\epsilon)\log^2 m)$ time. The algorithm works under the condition that the maximum area of overlap of A and B is at least some constant fraction of the area of A.

The algorithm is simple and follows the two-step framework of Section 4 in which an approximation of the best translation is followed by an approximation of the best rotation. However, now, the first step is a combination of grid sampling of the space of translations and random sampling of set A. This random sampling is based on the observation that the deterministic algorithm of Section 4 will compute a $(1-\epsilon)$-approximation k_{opt} times, where k_{opt} is the number of pairs of overlapping disks in an optimal solution. Intuitively, the larger this number is, the quicker such a pair will be tried out in the first step. Similar observations were made by Akutsu et al. [ATT98] who gave exact Monte Carlo algorithms for the largest common point set problem.

The second step is based on a direct application of the technique by Cheong et al. that allows us to maximize, up to an absolute error, the area of overlap under rotation in almost linear time, by computing a point of maximum depth in a one dimensional arrangement.

Rotations. For a given $\epsilon > 0$, we choose a uniform random sample S of points in A with $|S| = \Theta(\epsilon^{-2} \log m)$. For a point $s \in S$, we define $W(s) = \{\theta \in [0, 2\pi) | s(\theta) \in B\}$ where $s(\theta)$ denotes a copy of s rotated by θ. Let $\Theta_B(S)$ be the arrangement of all regions $W(s), s \in S$; it is a one-dimensional arragement of unions of rotational intervals.

Lemma 10 *Let θ_{opt} be the rotation that maximizes $\mathcal{V}(\theta)$. For any given $\epsilon > 0$, let S be a uniform random sample of points in A with $|S| \geq c_1 \frac{\log m}{\epsilon^2}$ where c_1 is an appropriate constant. A vertex θ_{apx} of $\Theta_B(S)$ of maximum depth satisfies $\mathcal{V}(\theta_{opt}) - \mathcal{V}(\theta_{apx}) \leq \epsilon V(A)$ with probability at least $1 - 1/m^6$.*

Proof: The proof is very similar to the proofs of Lemma 4.1 and Lemma 4.2 by Cheong et al. [CEHP04]. $\square$

$\Theta_B(S)$ has $O((m/\epsilon^2) \log m)$ complexity and can be computed in $O((m/\epsilon^2) \log(m/\epsilon) \log m)$ time by sorting. A vertex θ_{apx} of $\Theta_B(S)$ of maximum depth can be found by a simple traversal of this arrangement.

We could apply the idea above directly to rigid motions and compute the arrangement of all regions $W(s)$ with respect to rigid motions of S. Lemma 10 holds for this arrangement, and a vertex of maximum depth gives an absolute error on $\mathcal{V}(\vec{t}_{\text{opt}}, \theta_{\text{opt}})$. This arrangement has $O(|S|^3 m^2) = O((m^2/\epsilon^6) \log^3 m)$ vertices [CGH+97] that correspond — in workspace — to combinations of triples of points in S and triples of disks in B such that each point lies on the boundary of a disk. All such possible combinations can be easily found in $O((m^2/\epsilon^6) \log(m/\epsilon) \log^3 m)$ time. However, computing the actual rigid motion for any such combination is not trivial, as already explained in section 2. This complication is avoided by applying the technique to rotations only, thus computing a one-dimensional arrangement instead.

Rigid motions. Since we assume that $\mathcal{V}(\vec{t}_{\text{opt}}, \theta_{\text{opt}}) \geq \alpha V(A)$, for some given constant $0 < \alpha \leq 1$, we have that $k_{opt} \geq \alpha n$. Based also on the fact that the number of disks in A that participate in an optimal solution is at least $k_{opt}/6$, we can easily prove that the probability that $\Theta(\alpha^{-1} \log m)$ uniform random draws of disks from A will all fail to give a disk participating in an optimal solution is at most $1/m^6$. Algorithm RandomRigidMotion is given in Figure 5.

RandomRigidMotion(A, B, α, ϵ):

1. Choose a uniform random sample S of points in A, with $|S| = \Theta(\epsilon^{-2} \log m)$.
2. Let G be a uniform grid of spacing $c\epsilon$, where c is a suitable constant.
   Repeat $\Theta(\alpha^{-1} \log m)$ times:
   (a) Choose a random A_i from A.
   (b) For each $B_j \in B$ do:
      i. Set the center of rotation, i.e. the origin, to be B_j's center by translating B appropriately.
      ii. Let $T_{ij} = B_j \ominus A_i$, and determine all grid points $\vec{t}_g$ of G such that $\vec{t}_g \in T_{ij}$. For each such $\vec{t}_g$ do:
         - Compute a vertex θ^g_{apx} of maximum depth in $\Theta_B(S(\vec{t}_g))$, and $\mathcal{V}(\vec{t}_g, \theta^g_{apx})$.
3. Report the pair $(\vec{t}_{apx}, \theta_{apx})$ that maximizes $\mathcal{V}(\vec{t}_g, \theta^g_{apx})$.

Figure 5: Algorithm RandomRigidMotion(A, B, α, ϵ).

Theorem 11 *Let $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_m\}$, be two sets of disjoint unit disks in the plane and $I_{\vec{t}_{opt}, \theta_{opt}}$ be a rigid motion that maximizes $\mathcal{V}(\vec{t}, \theta)$. Assume that $\mathcal{V}(\vec{t}_{opt}, \theta_{opt}) \geq \alpha V(A)$, for some given constant $0 < \alpha \leq 1$. For any given $\epsilon > 0$,* RandomRigidMotion(A, B, α, ϵ) *computes a rigid motion $I_{\vec{t}_{apx}, \theta_{apx}}$ such that $\mathcal{V}(\vec{t}_{apx}, \theta_{apx}) \geq (1-\epsilon)\mathcal{V}(\vec{t}_{opt}, \theta_{opt})$ in $O((m^2/\epsilon^4)$ $\log(m/\epsilon)\log^2 m)$ time. The algorithm succeeds with probability at least $1 - 2/m^6$.*

Proof: Recall that $\bar{k}_{opt}$ is the number of disks A_i that participate in an optimal solution. Since $\bar{k}_{opt} > k_{opt}/6$, we have that $\mathbf{Pr}[(A_i \notin A_{opt})] < 1 - \frac{k_{opt}}{6n}$, for a random $A_i \in A$. The probability that all $|R_A|$ random draws from A will fail to give a disk that belongs to an optimal pair is

$$\mathbf{Pr}[R_A \cap A_{opt} = \emptyset] \leq (1 - \frac{k_{opt}}{6n})^{|R_A|} \leq e^{-k_{opt}|R_A|/(6n)} \leq e^{-\alpha|R_A|/6}.$$

By choosing $|R_A| \geq (36/\log e)\alpha^{-1}\log m$, we have that $\mathbf{Pr}[R_A \cap A_{opt} = \emptyset] \leq m^{-6}$.

Observe that if $R_A \cap A_{opt} \neq \emptyset$, then at least one intersecting pair $A_i(\vec{t}_{\rm opt}, \theta_{\rm opt}), B_j$ will be identified in the first loop. Then, in the second loop, the algorithm finds a $\theta^g_{\rm apx}$ for which, by Lemma 10, $\mathcal{V}(\vec{t}_g, \theta^g_{\rm opt}) - \mathcal{V}(\vec{t}_g, \theta^g_{\rm apx}) \leq \epsilon V(A)$, for some $\vec{t}_g \in T_{ij}$ with $|\vec{t}_{\rm opt} - \vec{t}_g| = O(\epsilon)$, and with probability at least $1 - m^{-6}$. As in the proof of Theorem 7, we have that $\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt}) - \mathcal{V}(\vec{t}_g, \theta_{\rm opt}) = O(k_{opt}\epsilon)$, $\mathcal{V}(\vec{t}_g, \theta_{\rm opt}) \leq \mathcal{V}(\vec{t}_g, \theta^g_{\rm opt})$ and $\mathcal{V}(\vec{t}_g, \theta^g_{\rm apx}) \leq \mathcal{V}(\vec{t}_{apx}, \theta_{\rm apx})$. Hence

$$\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt}) - \mathcal{V}(\vec{t}_{apx}, \theta_{apx}) = O(k_{opt}\epsilon) + \epsilon V(A).$$

Using that $\mathcal{V}(\vec{t}_{opt}, \theta_{opt}) \geq \alpha V(A)$ and $\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt}) = \Theta(k_{opt})$, the approximation bound follows. The algorithm fails to return such a pair $\vec{t}_{\rm apx}, \theta_{\rm apx}$ if and only if any of its two random sampling steps fail. That is, the algorithm fails with probability at most $2m^{-6}$.

Regarding the running time, the random sampling of set A can be easily done in $O((n/\epsilon^2)\log m)$ time. In the second step, for each of the $O(m/\epsilon^2)\log m$ grid translations $\vec{t}_g$, the one dimensional arrangement $\Theta_{A(\vec{t}_g)}(S)$ of $O((m/\epsilon^2)\log m)$ complexity is computed, a vertex of maximum depth $\theta^g_{\rm apx}$ is returned and $\mathcal{V}(\vec{t}_g, \theta^g_{\rm apx})$ is evaluated; this takes in $O((m/\epsilon^2)\log(m/\epsilon)\log m) + m\log m)$ time. In total the running time is $O((m^2/\epsilon^4)\log(m/\epsilon)\log^2 m)$. ◻

6 Sets of intersecting disks with different radii

We can generalize our results to the case where A and B consist of possibly intersecting and various size disks. Let r_s and r_l be the smallest, resp. largest disk radius among all disks in $A \cup B$. We define the *depth* of a point $p \in \mathbb{R}^2$ with respect to a set of disks as the number of disks in the set that contain it. Our algorithms work under the following two conditions: (i) $r_l/r_s = \rho$, for some constant $\rho > 0$; without loss of generality we assume that $r_s = 1$ and $r_l = \rho$, and (ii) the depth of any point $p \in \mathbb{R}^2$ with respect to A and the depth of any point $p \in \mathbb{R}^2$ with respect to B are both bounded by some constant, β.

First, we show that the assumptions result in denser sampling of configuration space with constants that depend on the parameters ρ and β as well. Then, we discuss their algorithmic implications.

Translations. First, consider Lemma 6. The maximum loss per pair is now determined by a pair of disks of radius ρ each: $\mathcal{V}_{ij}(\vec{t}_{\rm opt}, \theta_{\rm opt}) - \mathcal{V}_{ij}(\vec{t}, \theta_{\rm opt}) < 2\rho|\vec{t}_{opt} - \vec{t}| = O(\delta)$. Therefore, $\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt}) - \mathcal{V}(\vec{t}, \theta_{\rm opt}) < 2k_{opt}\rho|\vec{t}_{\rm opt} - \vec{t}| = O(k_{opt}\delta)$ and the lemma holds. Moreover, Theorem 2 holds as well, with the constant in the Θ-notation depending on both ρ and β.

Regarding the algorithm TRANSLATION, special care needs to be taken to avoid overcounting $\mathcal{V}(\vec{t}_g)$. We can do this in the following way: Consider the arrangement $\mathcal{A}$ of all disks $A_i \in A$ in the work space. Since the maximum depth in $\mathcal{A}$ is constant, $\mathcal{A}$ has $O(n)$ complexity and can be computed in $O(n \log n)$ time. Next, we compute a vertical decomposition $\mathcal{VD}(\mathcal{A})$ of $\mathcal{A}$; $\mathcal{VD}(\mathcal{A})$ has $O(n)$ disjoint cells[3] each of constant complexity and can be computed in $O(n \log n)$ time. Similarly, we compute $\mathcal{B}$ and $\mathcal{VD}(\mathcal{B})$ both in $O(m \log m)$ time. The loop in step 2 is now executed for every pair of cells $c_i \in \mathcal{VD}(\mathcal{A})$ and $c_j \in \mathcal{VD}(\mathcal{B})$ and instead of computing $\mathcal{V}_{ij}(\vec{t}_g)$, we compute $V(c_i(\vec{t}_g) \cap c_j)$. The voting scheme proceeds as before and runs within the same time bounds.

Rotations. Consider Lemma 3 and its proof: the length of sampling intervals is now determined by a pair of disks of radius ρ each. For such a pair A_i, B_j we have $\frac{\partial v(\theta)}{\partial \theta} \geq -2r_i\rho$. Therefore, for any pair of disks $A_i \in A$ and $B_j \in B$, we can sample $\mathcal{V}_{ij}(\theta)$ at regular intervals whose length is at most $\delta/(2r_i\rho)$ assuring that the loss per pair is at most 2δ. We also have to make sure that the number of samples per pair remains bounded, see Lemma 4. Indeed, $|R_{ij}|$ is maximized for the 'worst case' pair of disks of radius ρ each; this is a scaled, by ρ, version of the original problem.

Regarding algorithm ROTATION(A, B, δ), we use spacing of $\delta/(2r_i\rho)$ in its first step. Unfortunately, the simple technique used in the algorithm of Figure 3 to approximate $\mathcal{V}(\theta)$ for all the values $\theta \in \Theta$ does not work here since the disks in each set are possibly intersecting and the area of overlap accumulated in $\tilde{\mathcal{V}}(\theta)$ can be a bad approximation of $\mathcal{V}(\theta)$. We can overcome this problem in the following way. We compute $\mathcal{VD}(\mathcal{A})$ and $\mathcal{VD}(\mathcal{B})$ as before. Observe that every cell $c_i \in \mathcal{VD}(\mathcal{A})$ is fully contained in some disk in A; similarly, every cell $c_j \in \mathcal{VD}(\mathcal{B})$ is fully contained in some disk in B. Consider the function $V(c_i(\theta) \cap c_j)$, $\theta \in [0, 2\pi)$; for every pair (c_i, c_j), the error in $V(c_i(\theta) \cap c_j)$ is bounded by the error in the pair of their corresponing disks. Since each cell in both decompositions has at most two vertical walls and at most two circular segments, the function has a bounded number of local minima/maxima. We insert all these values, for every pair of cells, in set Θ. The algorithm proceeds as before by considering whether $V(c_i(\theta) \cap c_j)$ increases, decreases or reaches an optimum at each $\theta \in \Theta$. ROTATION now runs again in $O((mn/\delta) \log m)$ time and its correctness can be shown as in the proof of Lemma 5.

Rigid Motions. In addition to the relevant changes mentioned in the previous paragraphs, observe that a simple volume argument shows that any disk $A_i(\vec{t}, \theta)$ cannot intersect more than $9\rho^2\beta$ disks B_j for any $\vec{t}, \theta$. Thus, $|A_{opt}| > k_{opt}/(9\rho^2\beta)$.

In RIGIDMOTION, we compute $\mathcal{V}(\vec{t}_g, \theta^g_{apx})$ for each pair $(\vec{t}_g, \theta^g_{apx})$ in a straightforward way as follows: we compute $V(\bigcup A)$ in $O(n \log n)$ time, by computing $\mathcal{VD}(\mathcal{A})$ and summing up the areas of all its $O(n)$ cells. Similarly, we compute $V(\bigcup B)$ in $O(m \log m)$ time and, for each pair $(\vec{t}_g, \theta^g_{apx})$, $V((\bigcup A(\vec{t}_g, \theta^g_{apx})) \cup (\bigcup B))$ in $O(m \log m)$ time. It follows that for each pair $(\vec{t}_g, \theta^g_{apx})$ we can compute $V((\bigcup A(\vec{t}_g, \theta^g_{apx})) \cap (\bigcup B))$ in $O(m \log m)$ time. By incorporating all these changes, we can prove Theorem 7 as before.

Regarding the extension of Theorem 9, we apply the same method that we use in its proof, namely computing in ROTATION the disks $A_{i'}, B_{j'}$ such that $R_{i'j'}$ is not empty. Then, for each cell $c_{i'} \in A_{i'} \cap \mathcal{VD}(\mathcal{A})$ and each cell $c_{j'} \in B_{j'} \cap \mathcal{VD}(\mathcal{B})$ we proceed like before. We also need to keep track of the pairs $c_{i'}, c_{j'}$ that have been already added to avoid overcounting them. To show that the same time bound holds, we need to arguee that there are assymptotically the same number of pairs $c_{i'} \in \mathcal{VD}(\mathcal{A})$ and $c_{j'} \in \mathcal{VD}(\mathcal{B})$ with $R_{i'j'} \neq \emptyset$ than we had for the case of disjoint unit disks. For this, we first observe that each disk $A_{i'}$ is decomposed into $O(1)$ cells in $\mathcal{VD}(\mathcal{A})$ because it intersects at most $9\rho^2\beta$ other disks of A. The same holds for any $B_{j'}$. Therefore, each pair of disks $A_{i'}, B_{j'}$ that we need to consider, gives rise to $O(1)$ pairs of cells $c_{i'}, c_{j'}$.

[3]For our purpose, we only consider cells that are inside $\bigcup A$.

It remains to bound the number of pairs $A_{i'}, B_{j'}$ such that $R_{i',j'} \neq \emptyset$. For this, observe that set A can be decomposed into $O(1)$ disjoint groups of disjoint disks. This can be shown using a greedy procedure: compute a maximal set of disjoint disks $\tilde{A} \subset A$, that is, any disk in A intersects some disk in $\tilde{A}$; then take $\tilde{A}$ as a disjoint group and proceed recursively with $A \setminus \tilde{A}$. After $9\rho^2\beta + 1$ steps all the disks must be in some group, as any remaining disk must intersect a disk in each of the $9\rho^2\beta + 1$ groups, which is not possible. We can then apply Lemma 8 to each of the disjoint groups, and because there are a constant number of groups, we get the same assymptotic value for $\sum_i k_{j,j',i}$ as we had for the case of disjoint, unit disks. The result follows.

In RandomRigidMotion the size of R_A has to be at least $(54\rho^4\beta/\log e)\alpha^{-1}\log m$ since the condition $\mathcal{V}(\vec{t}_{opt}, \theta_{opt}) \geq \alpha V(A)$ now gives that $k_{opt} \geq \alpha n/\rho^2$. Note that Lemma 10 holds for any two planar regions A and B and thus for the two unions $\bigcup A$ and $\bigcup B$ as well. We can compute the sample points in A using $\mathcal{VD}(\mathcal{A})$. Last we compute each $W(s)$ by checking all disks in B in $O(m \log m)$ time. The running time of the algorithm stays the same and Theorem 11 can now be proven as before.

References

[AG99] H. Alt and L. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In J.R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 121–153. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 1999.

[ATT98] T. Akutsu, H. Tamaki, and T. Tokuyama. Distribution of distances and triangles in a point set and algorithms for computing the largest common point sets. *Discrete and Computational Geometry*, 20, 1998.

[CEHP04] O. Cheong, A. Efrat, and S. Har-Peled. On finding a guard that sees most and a shop that sells most. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms*, page to appear, 2004.

[CGH+97] P. Chew, M. Goodrich, D. P. Huttenlocher, K. Kedem, J. M. Kleinberg, and D. Kravets. Geometric pattern matching under euclidean motion. *Computational Geometry: Theory and Applications*, 7:113–124, 1997.

[dBGK+03] M. de Berg, P. Giannopoulos, C. Knauer, R. van Oostrum, and R. C. Veltkamp. The area of overlap of two unions of convex objects under translations. Technical Report UU-CS-2003-025, Institute of Information and Computing Sciences, Utrecht University, The Netherlands, 2003.

[GIMV03] M. Gavrilov, P. Indyk, R. Motwani, and S. Venkatasubramanian. Combinatorial and experimental methods for approximate point pattern matching. *Algorithmica*, 38(2):59–90, 2003.