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

Mark de Berg† Sergio Cabello‡ Panos Giannopoulos§¶ Christian Knauer‖
René van Oostrum** Remco C. Veltkamp**

Abstract

Let A and B be two sets of n resp. m disjoint unit disks in the plane, with $m \geq n$. We consider the problem of finding a translation or rigid motion of A that maximizes the total area of 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. We give deterministic $(1-\epsilon)$-approximation algorithms for translations and for rigid motions, which run in $O((nm/\epsilon^2)\log(m/\epsilon))$ and $O((n^2m^2/\epsilon^3)\log m))$ time, respectively. For rigid motions, we can also compute a $(1-\epsilon)$-approximation in $O((m^2n^{4/3}\Delta^{1/3}/\epsilon^3)\log n \log m)$ time, where Δ is the diameter of set A. Under the condition that the maximum area of overlap is at least a constant fraction of the area of A, we give a probabilistic $(1-\epsilon)$-approximation algorithm for rigid motions that runs in $O((m^2/\epsilon^4)\log^2(m/\epsilon)\log m)$ time and succeeds with high probability. Our results 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 Motions.

1 Introduction

Shape matching is a fundamental problem in computer vision: given two shapes A and B, one wants to determine how closely A resembles B, according to some distance measure between

*A preliminary version of this paper appeared in *Proc. 9th Scandinavian Workshop on Algorithm Theory*, pages 138–149, 2004.

†Faculty of Mathematics and Computing Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands; m.t.d.berg@TUE.nl. Partially supported by the Netherlands' Organization for Scientific Research (NWO) under project no. 639.023.301.

‡Department of Mathematics, IMFM, and Department of Mathematics, FMF, University of Ljubljana, Jadranska 19, SI-1000 Ljubljana, Slovenia; sergio.cabello@fmf.uni-lj.si. Research conducted at IICS, Utrecht University, The Netheralnds, and partially supported by the Cornelis Lely Stichting.

§Corresponding author.

¶Humbolt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, D-10099 Berlin, Germany; panos@informatik.hu-berlin.de. Research conducted at IICS, Utrecht University, The Netherlands, and partially supported by the Dutch Technology Foundation STW, project UIF 5055, and by a Marie Curie Scolarship of the EC programme 'Combinatorics, Geometry, and Computation', FU Berlin, HPMT-CT-2001-00282.

‖Institut für Informatik, Freie Universität Berlin, Takustraße 9, D-14195 Berlin, Germany; Christian.Knauer@inf.fu-berlin.de.

**Institute of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands; {rene, Remco.Veltkamp}@cs.uu.nl.

the shapes. Usually one is allowed to apply a transformation to A—such as a translation or a rigid motion—and the problem is to find the transformation of A that minimizes its distance to B.

The shape-matching problem has received a lot of attention, both in the computer-vision and computational-geometry community; see the surveys by Hagedoorn and Veltkamp [14] and Alt and Guibas [5]. Many results deal with the case where the shapes A and B are given as point sets. In this case one often uses the Hausdorff distance as a distance measure. The translation that minimizes the Hausdorff distance between A and B in two dimensions can be found in $O(nm(n+m)\alpha(nm)\log(n+m))$ time, where $|A| = n$ and $|B| = m$ [17]. When rotations are allowed as well, the optimum transformation can be found in $O((m+n)^6 \log(mn))$ time [16].

When A and B are polygons in the plane, one can base the distance measure on their boundaries. For example, one can try to find the translation that minimizes the Hausdorff distance between their boundaries [2] or the Fréchet distance between the boundaries [6]. Since these distance measures are based on the boundaries of the objects, they are fairly sensitive to noise. The *area of overlap* (or the *area the symmetric difference*) of two polygons is less sensitive to noise [10, 4] and therefore more appropriate for certain applications.

The function describing the area of overlap of two simple polygons under translations was first studied by Mount et al. [18]. They showed that this function is continuous and has $O((nm)^2)$ pieces, where n and m are the numbers of vertices of the polygons, with each piece being a polynomial of degree at most two. A representation of the function can be computed in $O((nm)^2)$ time. No algorithm is known that computes the translation that maximizes the area of overlap and does not compute the complete representation of the overlap function. One of the open problems mentioned by Mount et al. was to give efficient matching algorithms for objects with curved boundaries. De Berg et al. [10], gave an $O((n+m)\log(n+m))$ algorithm for determining the optimal translation for *convex* polygons; they also gave a constant-factor approximation algorithm. Finally, Alt et al. [4] gave a constant-factor approximation of the minimum area of symmetric difference of two convex shapes.

We study the following problem: given two sets A and B of disks in the plane, we wish to find a rigid motion that maximizes the 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 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.

A first application comes from weighted point-set matching. Consider a two- or three - dimensional shape that is reduced to a set of descriptive feature points, each of them having a weight that corresponds to its importance: the higher the weight, the more important the point for the total shape. For example, a curve or a contour can be reduced to a set of points of high curvature. Giannopoulos and Veltkamp [13] used transportation distances for measuring the similarity between two such weighted point sets. Alternatively, we can assign to each point a ball centered at it with radius relative to its weight. Thus, the two shapes are represented as unions of balls and a possible measure of their similarity is the area of overlap of the two unions.

Furthermore, any two- or three-dimensional shape can be efficiently approximated by a finite union of disks or balls—see, for example, the works by O'Rourke et al. [19] and

Amenta et al. [7]. Fournier et al. [20] also used the union of disks or spheres representation to interpolate between two shapes. Provided that, in addition, the assumptions (i) and (ii) above are satisfied for the approximating sets, our algorithms can be used to match a variety of shapes; however, the constant in assumption (i) may become large when the approximated objects have fine details.

Finally, both assumptions make perfect sense in molecular modelling with the hard sphere model [15]. Under this model the radii range of the spheres is fairly restricted and no center of a sphere can be inside another sphere; a simple packing argument shows that the latter implies assumption (ii). A related problem with applications in protein shape matching was examined by Agarwal et al. [1], who gave algorithms for minimizing the Hausdorff distance between two unions of discs or balls under translations.

Independently of this work, Cheong et al. [8] presented a general 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 $O((m/\epsilon^4)\log^3 m)$, 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 good enough for measuring the similarity of two shapes that are not too dissimilar.

Our contributions are the following. First, we show in Section 2 that the maximum number of combinatorially distinct translations of A with respect to B can be as high as $\Theta(n^2 m)$. When rotations are considered as well, the complexity is $O(n^3 m^2)$. Moreover, the function describing the area of overlap is quite complex, even for combinatorially equivalent placements. Therefore, the focus of our paper is on approximation algorithms. Next, we give a lower bound on the maximum area of overlap under translations, expressed in the number of pairs of disks that contribute to that area. This is a vital ingredient of almost all our algorithms.

In the remaining sections, we present our approximation algorithms. For the sake of clarity we describe the algorithms for the case of disjoint unit disks. It is not hard to adapt the algorithms to sets of disks satisfying assumptions (i) and (ii) above; the necessary changes can be found in Section 7. For any $\epsilon > 0$, our algorithms compute a $(1-\epsilon)$-approximation of the optimum overlap. For translations we give an algorithm that runs in $O((nm/\epsilon^2)\log(m/\epsilon))$ time. Compared to the result of Cheong et al. mentioned above, our algorithm is slower but has a better dependence on ϵ, it is deterministic and its error is always *relative*, even when the optimum is small. It also forms an ingredient to our algorithm for rigid motions, which runs in $O((n^2 m^2/\epsilon^3)\log m)$ time. If Δ is the diameter of set A the running time of the latter becomes $O((m^2 n^{4/3}\Delta^{1/3}/\epsilon^3)\log n \log m)$, 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 $O(n)$. Finally, for the case where the area of overlap is a given fraction α of the area of the union of A, we present a probabilistic algorithm for rigid motions that runs in $O((m^2/\alpha^3\epsilon^4)\log^2(m/\epsilon)\log m)$ time and succeeds with high probability.

Our algorithms 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 (for rigid motions). This way, we first achieve an absolute error on the optimum, which, we then argue, is also a relative error because of the lower bound on the overlap in terms of the number of overlapping pairs. The deterministic algorithms employ a 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 the translation space using a uniform grid and random sampling of both input sets.

2 Basic properties of the overlap function

We start by introducing some notation. Let $A = \{A_1, \dots, A_n\}$ and $B = \{B_1, \dots, B_m\}$, 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 $A(\theta) = \{A_1(\theta), \dots, A_n(\theta)\}$. Similarly, translated only versions of A are denoted by $A(\vec{t}) = \{A_1(\vec{t}), \dots, A_n(\vec{t})\}$. 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 $A(\vec{t}, \theta) = \{A_1(\vec{t}, \theta), \dots, A_n(\vec{t}, \theta)\}$ 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 $I_{\vec{t}_{\mathrm{opt}},\theta_{\mathrm{opt}}}$ 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. 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 $\{p_1 - p_2 : p_1 \in A, p_2 \in B\}$.

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.

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 placements of A with respect to B is $\Theta(n^2 m)$ under translations, and $O(n^3 m^2)$ under rigid motions.*

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 $\Theta(n^2 m)$. We first prove the upper bound and then give a lower bound example. In the next paragraph, we omit θ since it is fixed.

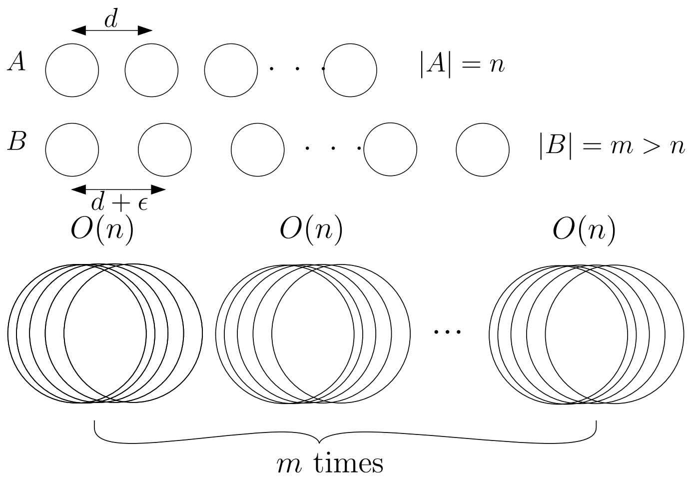

Figure 1: Two sets A and B and part of their configuration space $T(A, B)$ with highest $\Theta(n^2m)$ complexity.

Each set $B_j \ominus A_i$ is a disk of radius 2. Since the disks in both sets are closed and disjoint, no disk of one set can intersect more than five disks of the other set at any position. This is due to the 'kissing' or Hadwiger number of open unit disks, which is six [11]. This implies that a point in the configuration space cannot be covered by more than five of the m disks $B_j \ominus A_i$, for any fixed i, $1 \leq j \leq m$. In total, considering all values of i, no point in the total arrangement can be covered by more than $5n$ disks $B_j \ominus A_i$. A similar argument for the disks in B gives that no point can be covered by more than $5m$ disks $B_j \ominus A_i$. Thus, the maximum depth of any point in the arrangement is $5 \min\{m, n\} = 5n$. Since the complexity of the arrangement of n pseudodisks with maximum depth k is $O(nk)$ [21], the complexity of the configuration space is $O(n^2m)$.

To see that the bound is tight consider the sets A and B shown in Figure 1. Let d be the distance between the centers of any two consecutive disks in A and $d' = d + \epsilon$, $\epsilon \neq 0$, the distance between the centers of any two consecutive disks in B. These two sets give an arrangement of which the part of highest complexity is shown in Figure 1. In each of the $\Omega(m)$ 'bunches' of $\Theta(n)$ disks, the complexity is $\Theta(n^2)$, since each disk intersects all the others. In total, the complexity is $\Theta(n^2m)$.

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 disappear. 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. [9], the number of double events is $O(n^2m^2)$ and the number of triple events is $O(n^3m^2)$. Thus, the total number of cells created is $O(n^3m^2)$ and the complexity of the configuration space[1] is $O(n^3m^2)$ as well. ◻

This theorem implies that explicitly computing the subdivision of the configuration space

[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 $\theta \in [0, 2\pi)$.

into cells with combinatorially equivalent placements is highly expensive. Moreover, the computation for rigid motions is numerically unstable since it involves univariate equations of degree six [5]. 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. Therefore we turn our attention to approximation algorithms. The following theorem, which gives a lower bound on the maximum area of overlap, will be our fundamental took to convert an absolute error into a relative error.

Theorem 2 *Let $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_m\}$ be two sets of disjoint unit disks in the plane. Let $\vec{t}_{\mathrm{opt}}$ 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 $A_i(\vec{t}_{\mathrm{opt}}), B_j$, then $\mathcal{V}(\vec{t}_{\mathrm{opt}})$ is $\Theta(k_{\mathrm{opt}})$.*

Proof: First, note that $\mathcal{V}(\vec{t}_{\mathrm{opt}}) \leq k_{\mathrm{opt}}\pi$. Since we are considering only translations, the configuration space is two dimensional. For each pair $A_i(\vec{t}_{\mathrm{opt}}), B_j$ for which $A_i(\vec{t}_{\mathrm{opt}}) \cap B_j \neq \emptyset$, we draw, in configuration space, the region of translations $\mathcal{K}_{ij}$ that bring the center of $A_i(\vec{t}_{\mathrm{opt}})$ into B_j; see Figure 2. Such a region is a unit disk that is centered at a distance at most 2

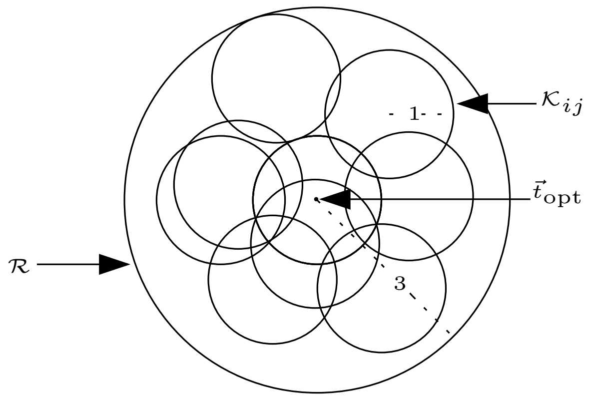

Figure 2: All disks $\mathcal{K}_{ij}$ are confined within a disk $\mathcal{R}$ of area 9π.

from $\vec{t}_{\mathrm{opt}}$. Thus, all regions $\mathcal{K}_{ij}$ are fully contained in a disk $\mathcal{R}$, centered at $\vec{t}_{\mathrm{opt}}$, of radius 3. By a simple volume argument, there must be a point $\vec{t}_\sharp \in \mathcal{R}$ (which represents a translation) that is covered by at least $k_{\mathrm{opt}}/9$ disks $\mathcal{K}_{ij}$. Each of the corresponding pairs $A_i(\vec{t}_\sharp), B_j$ has an overlap of at least $2\pi/3 - \sqrt{3}/2$. Thus, $\mathcal{V}(\vec{t}_{\mathrm{opt}}) \geq \mathcal{V}(\vec{t}_\sharp) \geq (2\pi/27 - \sqrt{3}/18)k_{\mathrm{opt}}$. □

3 Approximation algorithms for translations

Theorem 2 suggests the following simple approximation algorithm: compute the arrangement of the regions $\mathcal{K}_{ij}$, with $i = 1, \ldots, n$ and $j = 1, \ldots, m$, and pick any point of maximum depth. Such a point corresponds to a translation $\vec{t}$ that gives a constant-factor approximation. An analysis similar to that of Theorem 2 leads to a factor of $(2/3 - \sqrt{3}/2\pi) \approx 0.39$. It is possible to do much better, however.

We present a deterministic $(1-\epsilon)$-approximation algorithm; see Cheong et al.'s [8] for a probabilistic $(1-\epsilon)$-approximation algorithm. Since both A and B consist of disjoint disks, we have $\mathcal{V}(\vec{t}) = \sum_{A_i \in A, B_j \in B} \mathcal{V}_{ij}(\vec{t})$. The algorithm is based on sampling the configuration space by using a uniform grid. This is possible due to the following lemma, which tells that, in terms of absolute error, it is not too bad if we choose a translation which is close to the optimal one. As mentioned before, we can then use Theorem 2 to convert it to a relative error.

Lemma 3 *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 largest *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] , $\mathcal{V}(\vec{t}, \theta) - \mathcal{V}(\vec{t'}, \theta) = O(k\delta)$. ◻

The computation of $\mathcal{V}(\vec{t})$ for every grid translation is not done directly, but instead we use a voting scheme, which speeds up the algorithm by a linear factor. Algorithm TRANSLATION(A, B, ϵ) is shown in Figure 3.

TRANSLATION(A, B, ϵ):

1. Initialize an empty binary search tree $\mathcal{S}$ with entries of the form $(\vec{t}, \mathcal{V}(\vec{t}))$ where $\vec{t}$ is the key. Let G be a uniform grid of spacing $c\epsilon$, where c is a suitable constant.
2. For each pair of disks $A_i \in A$ and $B_j \in B$ do:
   (a) Determine all grid points $\vec{t}_{\mathrm{g}}$ of G such that $\vec{t}_{\mathrm{g}} \in T_{ij}$, where $T_{ij} = B_j \ominus A_i$. For each such $\vec{t}_{\mathrm{g}}$ do:
      - If $\vec{t}_{\mathrm{g}}$ is in $\mathcal{S}$, then $\mathcal{V}(\vec{t}_{\mathrm{g}}) := \mathcal{V}(\vec{t}_{\mathrm{g}}) + \mathcal{V}_{ij}(\vec{t}_{\mathrm{g}})$ otherwise, insert $\vec{t}_{\mathrm{g}}$ in $\mathcal{S}$ with $\mathcal{V}(\vec{t}_{\mathrm{g}}) := \mathcal{V}_{ij}(\vec{t}_{\mathrm{g}})$.
3. Report the grid point $\vec{t}_{\mathrm{apx}}$ that maximizes $\mathcal{V}(\vec{t}_{\mathrm{g}})$.

Figure 3: Algorithm TRANSLATION(A, B, ϵ).

Theorem 4 *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 $n \leq m$. Let $\vec{t}_{\mathrm{opt}}$ be a translation that maximizes $\mathcal{V}(\vec{t})$. Then, for any given $\epsilon > 0$,* TRANSLATION(A, B, ϵ) *computes a translation $\vec{t}_{\mathrm{apx}}$, for which $\mathcal{V}(\vec{t}_{\mathrm{apx}}) \geq (1-\epsilon)\mathcal{V}(\vec{t}_{\mathrm{opt}})$, in $O((mn/\epsilon^2)\log(m/\epsilon))$ time.*

Proof: It follows from Theorem 2 that there must be at least one pair of disks with significant overlap in an optimal translation. This implies that the grid point closest to the optimum must have a pair of overlapping disks, and so the algorithm checks at least one grid translation $\vec{t}_{\mathrm{g}}$ for which $|\vec{t}_{\mathrm{opt}} - \vec{t}_{\mathrm{g}}| = O(\epsilon)$. Let k_{opt} be the number of overlapping pairs $A_i(\vec{t}_{\mathrm{opt}}), B_j$. According

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

to Lemma 3, by setting $\theta = 0$, we have that $\mathcal{V}(\vec{t}_{\text{opt}}) - \mathcal{V}(\vec{t}_{\text{apx}}) \leq \mathcal{V}(\vec{t}_{\text{opt}}) - \mathcal{V}(\vec{t}_{\text{g}}) = O(k_{\text{opt}}\epsilon)$. By Theorem 2, we have $\mathcal{V}(\vec{t}_{\text{opt}}) = \Theta(k_{\text{opt}})$, and the approximation bound follows.

The algorithm considers $O(1/\epsilon^2)$ grid translations $\vec{t}_{\text{g}}$ per pair of disks. Each translation is handled in $O(\log(nm/\epsilon^2))$ time. Thus, the total running time is $O((nm/\epsilon^2)\log(nm/\epsilon^2)) = O((nm/\epsilon^2)\log(m/\epsilon))$. □

4 The rotational case

This section considers the following restricted scenario: set B is fixed, and set A can be rotated about 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 5 *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 4. 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

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

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

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

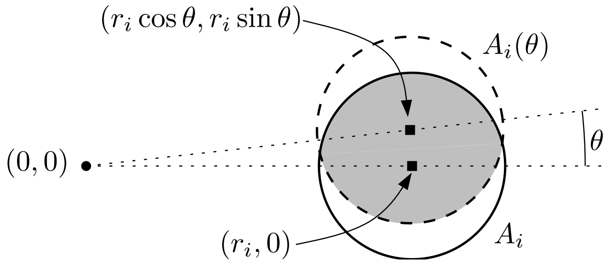

Figure 4: Notation in Lemma 5. The area of the grey region corresponds to $v(\theta)$.

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 6 *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

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

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 $\theta_{ij} = \arccos\frac{r_i^2+r_j^2-4}{2r_ir_j}$.

As shown in Figure 5, 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

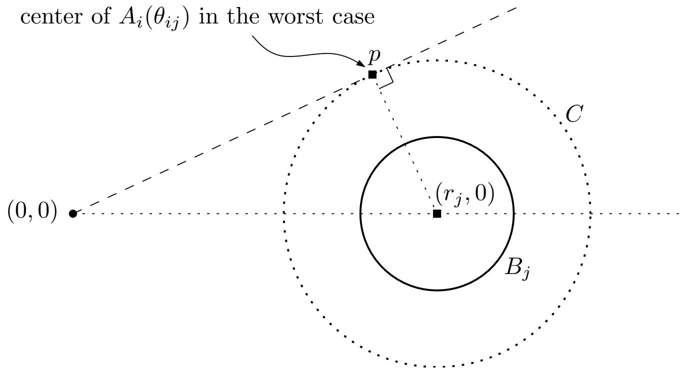

Figure 5: Notation in Lemma 6. The center of $A_i(\theta_{ij})$ is placed in the circle C. Therefore, θ_{ij} is maximized for the dashed line through the origin and tangent to C.

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

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

Using L'Hôpital's rule we can compute that

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

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

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 5 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. This is the idea for algorithm Rotation, described in detail in Figure 6; the value $\tilde{\mathcal{V}}(\theta)$ is the conservative estimate of $\mathcal{V}(\theta)$, as just explained.

Lemma 7 *Let θ_{opt} be a rotation that maximizes $\mathcal{V}(\theta)$ and let k_{opt} be the number of overlapping pairs $A_i(\theta_{\text{opt}}), B_j$. For any given $\delta > 0$, the rotation θ_{apx} reported by* Rotation*(A, B, δ) satisfies $\mathcal{V}(\theta_{\text{opt}}) - \mathcal{V}(\theta_{\text{apx}}) = O(k_{\text{opt}}\delta)$, and can be computed in $O(m \log n + (|\Pi|/\delta) \log m)$ time, where Π is the set of pairs A_i, B_j with $R_{i,j} \neq \emptyset$.*

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. It is important that the estimation $\mathcal{V}(\theta)$ is from below: since k_θ could be much larger than k_{opt}, an estimation from above with an error of $k_\theta\delta$ could be very large.

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 5 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 $B_j \in B$ with $R_{ij} \neq \emptyset$, choose a set $\Theta_{ij} := \{\theta_{ij}^1, \ldots, \theta_{ij}^{s_{ij}}\}$ 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 $\theta_0, \theta_1, \ldots$ 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 $\theta_{\text{apx}} \in \Theta$ that maximizes $\tilde{\mathcal{V}}(\theta)$.

Figure 6: 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. $A_M(\theta_{\text{opt}}) \cap (\bigcup B) \neq \emptyset$. Let $\tilde{\theta} \in \Theta$ be the closest value to θ_{opt}. We have

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

for all A_i in the optimal solution. Again, according to Lemma 5, the loss per pair A_i, B_j is $\mathcal{V}_{ij}(\theta_{\text{opt}}) - \mathcal{V}_{ij}(\tilde{\theta}) \leq 2\delta$. In total, $\mathcal{V}(\theta_{\text{opt}}) - \mathcal{V}(\tilde{\theta}) \leq 2k_{\text{opt}}\delta$.

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, it holds that $k_{\tilde{\theta}} = k_{\text{opt}}$.

Putting it all together we get

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

The running time of the algorithm is bounded as follows. First, we discuss how to construct the set Π, that is, pairs A_i, B_j with $R_{ij} \neq \emptyset$. We store the disks of A in a balanced search tree

T sorted by the distance from their center to the origin. This takes $O(n \log n)$ time. Consider a fixed disk B_j, and let Π_j be the pairs A_i, B_j in Π. Note that $R_{ij} \neq \emptyset$ if and only if the distance from the origin to the centers of A_i and B_j differs by at most two. Therefore, we can use T to construct Π_j in $O(|\Pi_j| + \log n)$ time. By considering each B_j, we construct Π in $O(n \log n) + O(\sum_j |\Pi_j| + m \log n) = O(\Pi + m \log n)$ time.

Once we have Π, we can construct the set Θ, which consists of $|\Pi|$ subsets Θ_{ij}, each with $O(1/\delta)$ rotations by Lemma 6. 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 $O((|\Pi|/\delta) \log |\Pi|) = O((|\Pi|/\delta) \log m)$ time. ◻

5 Approximation algorithms for rigid motions

Any rigid motion can be described as a translation followed by a rotation around the origin. This is used in algorithm RigidMotion described in Figure 7, which combines the algorithms for translations and for rotations to obtain a deterministic $(1-\epsilon)$-approximation for rigid motions.

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 $\vec{t}_{\mathrm{g}}$ of G such that $\vec{t}_{\mathrm{g}} \in T_{ij}$. For each such $\vec{t}_{\mathrm{g}}$ do:
      - run Rotation$(A(\vec{t}_{\mathrm{g}}), B, c'\epsilon)$, where c' is an appropriate constant. Let $\theta^{\mathrm{g}}_{\mathrm{apx}}$ be the rotation returned. Compute $\mathcal{V}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{apx}})$.
2. Report the pair $(\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}})$ that maximizes $\mathcal{V}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{apx}})$.

Figure 7: Algorithm RigidMotion(A, B, ϵ).

Theorem 8 *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 $I_{\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}}$ 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 $I_{\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}}$ such that $\mathcal{V}(\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}) \geq (1-\epsilon)\mathcal{V}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$ in $O((n^2 m^2/\epsilon^3) \log m)$ time.*

Proof: We will show that $\mathcal{V}(\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}})$ approximates $\mathcal{V}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$ 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 $|A_{\mathrm{opt}}| = \bar{k}_{\mathrm{opt}}$. Since the 'kissing' number of unit open disks is six, we have that $k_{\mathrm{opt}} < 6\bar{k}_{\mathrm{opt}}$, where k_{opt} is the number of overlapping pairs in the optimal solution. Next, imagine that RigidMotion$(A_{\mathrm{opt}}, B, \epsilon)$ is run instead of RigidMotion(A, B, ϵ). Of course, an optimal rigid motion for $A_{\rm opt}$ is an optimal rigid motion for A and the error we make by applying a non-optimal rigid motion to $A_{\rm opt}$ bounds the error we make when applying the same rigid motion to A.

Consider a disk $A_i \in A_{\rm opt}$ and an intersecting pair $A_i(\vec{t}_{\rm opt}, \theta_{\rm opt})$, B_j. Since, at some stage, the algorithm will use B_j's center as the center of rotation, and $I_{\vec{t}_{\rm opt},\theta_{\rm opt}} = R_{\theta_{\rm opt}} \circ T_{\vec{t}_{\rm opt}}$, we have that $A_i(\vec{t}_{\rm opt}) \cap B_j \neq \emptyset$ if and only if $A_i(\vec{t}_{\rm opt}, \theta_{\rm opt}) \cap B_j \neq \emptyset$. Hence, we have that $\vec{t}_{\rm opt} \in T_{ij}$ and the algorithm will consider some grid translation $\vec{t}_{\rm g} \in T_{ij} = B_j \ominus A_i$, for which $|\vec{t}_{\rm opt} - \vec{t}_{\rm g}| = O(\epsilon)$. By Lemma 3 we have $\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt}) - \mathcal{V}(\vec{t}_{\rm g}, \theta_{\rm opt}) = O(k_{\rm opt}\epsilon) = O(\bar{k}_{\rm opt}\epsilon)$.

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

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

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

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, $\mathcal{V}(\vec{t}_{\rm opt}, \theta_{\rm opt}) = \Theta(k_{\rm opt})$ and the approximation bound follows.

Finally, the running time of the algorithm is dominated by its first step. We can compute $\mathcal{V}(\vec{t}_{\rm g}, \theta^{\rm g}_{\rm apx})$ 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 $O(m + 1/\epsilon^2 + (1/\epsilon^2)(\Pi/\epsilon)\log m + (1/\epsilon^2)m \log m) = O((nm/\epsilon^3)\log m)$ time since $\Pi = O(nm)$. The step is executed nm times, thus the algorithm runs in $O((n^2m^2/\epsilon^3)\log m)$ time. ◻

5.1 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. [12]. 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 9 *[12, 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 $\Delta = O(n)$ [12], 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 10 *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 Δ be the diameter of A, and let $I_{\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}}$ be the rigid motion maximizing $\mathcal{V}(\vec{t}, \theta)$. For any $\epsilon > 0$, we can find in $O((m^2 n^{4/3} \Delta^{1/3}/\epsilon^3) \log n \log m)$ time a rigid motion $I_{\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}}$ such that $\mathcal{V}(\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}) \geq (1-\epsilon)\mathcal{V}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$.*

Proof: For indices i, j, let Z_{ij} be the set of pairs $A_{i'}, B_{j'}$ such that

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

where c_{A_i} denotes the center of A_i and c_{B_j} denotes the center of B_j. If A has diameter Δ, then $\sum_{i,j} |Z_{ij}| = O(m^2 n^{4/3} \Delta^{1/3} \log n)$: for each of the $O(m^2)$ pairs j, j', there are $O(n^{4/3} \Delta^{1/3} \log n)$ pairs i, i' because of Lemma 9.

Consider one of the calls to ROTATION that RIGIDMOTION makes. The origin is set at the center of some B_j, and some A_i intersects B_j. The pairs of disks $A_{i'}, B_{j'}$ with $R_{i'j'} \neq \emptyset$ is a subset of $Z_{i,j}$, and from Lemma 7 we conclude that the call to ROTATION takes $O(m \log n + (|Z_{ij}|/\epsilon) \log m)$ time. For each pair of disks A_i, B_j, the algorithm RIGIDMOTION makes $O(1/\epsilon^2)$ calls to ROTATION. Therefore, the total running time can be bounded by

$$\begin{aligned} & O\left(\sum_{i,j} O(1/\epsilon^2)\, O(m \log n + (|Z_{ij}|/\epsilon) \log m) \right) \\ &= O\left(nm^2 \log n/\epsilon^2 + (\epsilon^{-3} \log m) \sum_{i,j} |Z_{ij}|) \right) \\ &= O\left(nm^2 \log n/\epsilon^2 + (\epsilon^{-3} \log m) m^2 n^{4/3} \Delta^{1/3} \log n \right) \\ &= O((m^2 n^{4/3} \Delta^{1/3}/\epsilon^3) \log n \log m). \end{aligned}$$

□

6 A Monte Carlo algorithm for rigid motions

In this section we present a Monte Carlo algorithm that computes a $(1-\epsilon)$-approximation for rigid motions in $O((m^2/\alpha^3\epsilon^4) \log^2(m/\epsilon) \log m)$ time, where α is an input parameter that should have the property that the maximum area of overlap is at least $\alpha V(A)$.

The algorithm is simple and follows the two-step framework of Section 5 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 5 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. [3] who gave exact Monte Carlo algorithms for the largest common point set problem.

The second step is 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. Let S be a set of points from $\bigcup A$. For each 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 $\mathcal{A}_B(S)$ be the arrangement of all regions $W(s), s \in S$; it is a one-dimensional arrangement of unions of rotational intervals. The following lemma follows by a direct modification in the proof of Lemma 4.2 in Cheong et al. [8].

Lemma 11 *Let θ_{opt} be the rotation that maximizes $\mathcal{V}(\theta)$. For any given $\delta > 0$, let S be a uniform random sample of points in $\bigcup A$ with $|S| \geq c_1 \delta^{-2} \log(m/\delta)$, where c_1 is an appropriate constant. A vertex θ_{apx} of $\mathcal{A}_B(S)$ of maximum depth satisfies $|\mathcal{V}(\theta_{\text{opt}}) - \mathcal{V}(\theta_{\text{apx}})| \leq \delta V(A)$ with probability at least $1 - \delta^2/m^6$.*

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

We could apply the idea above directly to rigid motions, with $\delta = \alpha\epsilon$, and consider the three-dimensional regions $W(s)$ with respect to rigid motions of S. Lemma 11 holds for the arrangement of all these regions and a vertex of maximum depth gives an absolute error on $\mathcal{V}(\vec{t}_{\text{opt}}, \theta_{\text{opt}})$. Computing this arrangement explicitly is quite cumbersome since each $W(s)$ is bounded by curved surfaces. However, one need not compute the whole arrangement, but rather test all its possible vertices. Chew et al. [9] showed that this arrangement has $O(|S|^3 m^2) = O((m^2/\delta^6)\log^3(m/\delta))$ vertices 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. They also showed that all such possible combinations can be found in $O((m^2/\delta^6)\log^4(m/\delta))$ time using dynamic Voronoi diagrams. However, computing the actual rigid motion for any such combination is not trivial, as it requires solving algebraic equations of degree six. By applying the technique to rotations only, thus computing a one-dimensional arrangement, we avoid this complication in addition to achieving a better dependence on ϵ.

Rigid motions. Since we assume that $\mathcal{V}(\vec{t}_{\text{opt}}, \theta_{\text{opt}}) \geq \alpha V(A)$, for some given value $0 < \alpha \leq 1$, we have that $k_{\text{opt}} \geq \alpha n$. Based also on the fact that the number of disks in A that participate in an optimal solution is at least $k_{\text{opt}}/6$, 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 $1/m^5$. This, together with Lemma 11 is the main idea used in Algorithm RandomRigidMotion, given in Figure 8.

Theorem 12 *Let $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_m\}$, with $m \leq n$ be two sets of disjoint unit disks in the plane, and let $I_{\vec{t}_{\text{opt}}, \theta_{\text{opt}}}$ be a rigid motion that maximizes $\mathcal{V}(\vec{t}, \theta)$. Assume that $\mathcal{V}(\vec{t}_{\text{opt}}, \theta_{\text{opt}}) \geq \alpha V(A)$, for some given value $0 < \alpha \leq 1$. For any given $\epsilon > 0$,* RandomRigidMotion(A, B, α, ϵ) *computes a rigid motion $I_{\vec{t}_{\text{apx}}, \theta_{\text{apx}}}$ such that $\mathcal{V}(\vec{t}_{\text{apx}}, \theta_{\text{apx}}) \geq (1-\epsilon)\mathcal{V}(\vec{t}_{\text{opt}}, \theta_{\text{opt}})$ in $O((m^2/\alpha^3\epsilon^4)\log^2(m/\epsilon)\log m)$ time. The algorithm succeeds with probability $1 - O(m^{-5})$.*

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

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

RandomRigidMotion(A, B, α, ϵ):

1. Choose a uniform random sample S of points in $\bigcup A$, with $|S| = \Theta(\delta^{-2} \log(m/\delta))$, where $\delta = \alpha\epsilon$.
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 $\vec{t}_{\mathrm{g}}$ of G such that $\vec{t}_{\mathrm{g}} \in T_{ij}$. For each such $\vec{t}_{\mathrm{g}}$ do:
         - Compute a vertex $\theta^{\mathrm{g}}_{\mathrm{apx}}$ of maximum depth in $\mathcal{A}_B(S(\vec{t}_{\mathrm{g}}))$, and $\mathcal{V}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{apx}})$.
3. Report the pair $(\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}})$ that maximizes $\mathcal{V}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{apx}})$.

Figure 8: Algorithm RandomRigidMotion(A, B, α, ϵ).

By choosing $|R_A| \geq (30/\log e)\alpha^{-1}\log m$, we have that $\mathbf{Pr}[R_A \cap A_{\mathrm{opt}} = \emptyset] \leq m^{-5}$.

For bounding the error in the approximation, we assume from now on that $R_A \cap A_{\mathrm{opt}} \neq \emptyset$. For each translation $\vec{t}_{\mathrm{g}}$ that the algorithm tries in step 2(b)ii, Lemma 11 implies that $|\mathcal{V}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{opt}}) - \mathcal{V}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{apx}})| \leq \delta V(A) = \alpha\epsilon V(A)$, with probability at least $1 - 1/m^6$. Since we are using $O(m/\alpha\epsilon^2)$ times Lemma 11, the probability of error in any of them is bounded by $O(m/\alpha\epsilon^2)(\delta^2/m^6) = O(m^{-5})$. From now on, we also assume that all uses of Lemma 11 are correct.

Under the assumption that $R_A \cap A_{\mathrm{opt}} \neq \emptyset$, the algorithm will try one intersecting pair $A_i(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}), B_j$ in the first loop, and, in particular, it will try a translation $\vec{t}_{\mathrm{g}}$ such that $|\vec{t}_{\mathrm{opt}} - \vec{t}_{\mathrm{g}}| \leq c\epsilon$. As in Theorem 8, we have that $\mathcal{V}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}) - \mathcal{V}(\vec{t}_{\mathrm{g}}, \theta_{\mathrm{opt}}) = O(k_{\mathrm{opt}}\epsilon)$, $\mathcal{V}(\vec{t}_{\mathrm{g}}, \theta_{\mathrm{opt}}) \leq \mathcal{V}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{opt}})$ and $\mathcal{V}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{apx}}) \leq \mathcal{V}(\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}})$. Hence

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

Using that $\mathcal{V}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}) = \Theta(k_{\mathrm{opt}})$, the approximation bound follows. The algorithm fails to return such a pair $\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}$ if and only if it fails to obtain $R_A \cap A_{\mathrm{opt}} \neq \emptyset$ or any of the uses of Lemma 11 fails. From the discussion above, it follows that this happens with probability $O(m^{-5})$.

As for the running time, the random sampling of set A can be done in $O((n/\delta^2)\log(m/\delta)) = O((n/\alpha^2\epsilon^2)\log(m/\epsilon))$ time, where we have used that $\alpha \geq 1/n$. In the second step, for each of the $O((m/\alpha\epsilon^2)\log m)$ grid translations $\vec{t}_{\mathrm{g}}$, we spend $O((m/\delta^2)\log^2(m/\delta)) = O((m/\alpha^2\epsilon^2))\log^2(m/\epsilon))$ time to construct the arrangement $\mathcal{A}_B(S(\vec{t}_{\mathrm{g}}))$, to find a vertex of maximum depth $\theta^{\mathrm{g}}_{\mathrm{apx}}$, and to evaluate $\mathcal{V}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{apx}})$. Therefore, we spend $O((m^2/\alpha^3\epsilon^4)\log^2(m/\epsilon)\log m)$ time in total. ◻

7 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 3. The maximum loss per pair is now determined by a pair of disks of radius ρ each: $\mathcal{V}_{ij}(\vec{t}_{\text{opt}}, \theta_{\text{opt}}) - \mathcal{V}_{ij}(\vec{t}, \theta_{\text{opt}}) < 2\rho|\vec{t}_{\text{opt}} - \vec{t}| = O(\delta)$. Therefore, $\mathcal{V}(\vec{t}_{\text{opt}}, \theta_{\text{opt}}) - \mathcal{V}(\vec{t}, \theta_{\text{opt}}) < 2k_{\text{opt}}\rho|\vec{t}_{\text{opt}} - \vec{t}| = O(k_{\text{opt}}\delta)$ 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 $\mathcal{V}(\vec{t}_{\text{g}})$. 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 $\mathcal{V}_{ij}(\vec{t}_{\text{g}})$, we compute $V(c_i(\vec{t}_{\text{g}}) \cap c_j)$. The voting scheme proceeds as before and runs within the same time bounds.

Rotations. Consider Lemma 5 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 6. 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 6 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 corresponding 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. Also, each disk

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

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 disk B_j in $\mathcal{VD}(\mathcal{B})$. Hence, the total number of the additional values in Θ is of $O(\Pi)$. 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 $O((\Pi/\delta)\log m)$ time and its correctness can be shown as in the proof of Lemma 7.

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, $|A_{\text{opt}}| > k_{\text{opt}}/(9\rho^2\beta)$.

In RigidMotion, we compute $\mathcal{V}(\vec{t}_{\text{g}}, \theta^{\text{g}}_{\text{apx}})$ for each pair $(\vec{t}_{\text{g}}, \theta^{\text{g}}_{\text{apx}})$ 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 $(\vec{t}_{\text{g}}, \theta^{\text{g}}_{\text{apx}})$, $V((\bigcup A(\vec{t}_{\text{g}}, \theta^{\text{g}}_{\text{apx}})) \cup (\bigcup B))$ in $O(m \log m)$ time. It follows that for each pair $(\vec{t}_{\text{g}}, \theta^{\text{g}}_{\text{apx}})$ we can compute $V((\bigcup A(\vec{t}_{\text{g}}, \theta^{\text{g}}_{\text{apx}})) \cap (\bigcup B))$ in $O(m \log m)$ time. By incorporating all these changes, we can prove Theorem 8 as before.

Regarding the extension of Theorem 10, 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 argue that there are asymptotically the same number of pairs $c_{i'} \in \mathcal{VD}(\mathcal{A})$ and $c_{j'} \in \mathcal{VD}(\mathcal{B})$ with $R_{i'j'} \neq \emptyset$ as we had for the case of disjoint unit disks. Since each disk $A_{i'}$ is decomposed into $O(1)$ cells in $\mathcal{VD}(\mathcal{A})$, and the same holds for any $B_{j'}$ in $\mathcal{VD}(\mathcal{B})$, each pair of disks $A_{i'}, B_{j'}$ that we need to consider gives rise to $O(1)$ pairs of cells $c_{i'}, c_{j'}$.

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. Then, we can apply Lemma 9 to each of the disjoint groups, and since there are a constant number of groups, we get the same asymptotic running time as in the case of disjoint, unit disks.

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 $\mathcal{V}(\vec{t}_{\text{opt}}, \theta_{\text{opt}}) \geq \alpha V(A)$ now gives that $k_{\text{opt}} \geq \alpha n/\rho^2$. Note that Lemma 11 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 12 can now be proven as before.

8 Concluding remarks

We have presented approximation algorithms for the maximum area of overlap of two sets of disks in the plane. Theorem 2 on the lower bound on the maximum area of overlap generalizes to three dimensions in a straightforward way. The approximation algorithm for translations generalizes as well, in the following way: the arrangement of n spheres (under the assumptions (i) and (ii) of Section 1) has $O(n)$ complexity and can be computed in $O(n \log n)$ time [15].

In addition, there exists a decomposition of this arrangement into $O(n)$ simple cells that can be computed in $O(n \log n)$ time [15]. By using these cells in the voting scheme, the running time of the algorithm is $O((mn/\epsilon^3) \log(mn/\epsilon))$.

Although our algorithms for rigid motions generalize to 3D, their running times increase dramatically. It would be worthwhile to study this case in detail, refine our ideas and give more efficient algorithms.

References

[1] P. K. Agarwal, S.Har-Peled, M. Sharir, and Y. Wang. Hausdorff distance under translation for points, disks, and balls. In *Proc. 19th Annu. ACM Sympos. Comput. Geom.*, pages 282–291, 2003.

[2] P. K. Agarwal, M. Sharir, and S. Toledo. Applications of parametric searching in geometric optimization. *J. Algorithms*, 17:292–318, 1994.

[3] 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 Comput. Geom.*, 20, 1998.

[4] H. Alt, U. Fuchs, G. Rote, and G. Weber. Matching convex shapes with respect to the symmetric difference. *Algorithmica*, 21:89–103, 1998.

[5] 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.

[6] H. Alt, C. Knauer, and C. Wenk. Matching polygonal curves with respect to the Fréchet distance. In *Proc. 18th Internat. Sympos. Theoretical Aspects of Computer Science*, volume 2010 of *Lecture Notes in Computer Science*, pages 63–74, 2001.

[7] N. Amenta and R. Kolluri. Accurate and efficient unions of balls. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 119–128, 2000.

[8] O. Cheong, A. Efrat, and S. Har-Peled. On finding a guard that sees most and a shop that sells most, 2005. Accepted for publication in *Discrete Comput. Geom.* and available as `http://tclab.kaist.ac.kr/~otfried/Papers/ceh-fgsmssm.pdf`. A preliminary version appeared in *Proc. 15th ACM-SIAM Sympos. Discrete Algorithms,* pages 1098–1107, 2004.

[9] P. Chew, M. Goodrich, D. P. Huttenlocher, K. Kedem, J. M. Kleinberg, and D. Kravets. Geometric pattern matching under euclidean motion. *Comput. Geom. Theory Appl.*, 7:113–124, 1997.

[10] M. de Berg, O. Devillers, M. van Kreveld, O. Schwarzkopf, and M. Teillaud. Computing the maximum overlap of two convex polygons under translations. *Theory Comput. Systems*, 31:613–628, 1998.

[11] G. Fejes Tóth. Packing and covering. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 2, pages 19–42. CRC Press LLC, Boca Raton, FL, 1997.

[12] M. Gavrilov, P. Indyk, R. Motwani, and S. Venkatasubramanian. Combinatorial and experimental methods for approximate point pattern matching. *Algorithmica*, 38(1):59–90, 2003.

[13] P. Giannopoulos and R. C. Veltkamp. A pseudo-metric for weighted point sets. In *Proc. 7th European Conf. Computer Vision*, volume 2352 of *Lecture Notes in Computer Science*, pages 715–731. Springer, 2002.

[14] M. Hagedoorn and R. C. Veltkamp. State-of-the-art in shape matching. In M. Lew, editor, *Principles of Visual Information Retrieval*, pages 87–119. Springer, 2001.

[15] D. Halperin and M. H. Overmars. Spheres, molecules, and hidden surface removal. *Comput. Geom. Theory Appl.*, 11(2):83–102, 1998.

[16] D. P. Huttenlocher, K. Kedem, and J. M. Kleinberg. On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under euclidean motion in the plane. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 110–120, 1992.

[17] D. P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of Voronoi surfaces and its applications. *Discrete Comput. Geom.*, 9:267–291, 1993.

[18] D. M. Mount, R. Silverman, and A. Y. Wu. On the area of overlap of translated polygons. *Computer Vision and Image Understanding*, 64:53–61, 1996.

[19] J. O'Rourke and N. Badler. Decomposition of three dimensional objects into spheres. *IEEE Trans. Pattern Analysis Machine Intelligence*, 1(3):295–305, July 1979.

[20] V. Ranjan and A. Fournier. Matching and interpolation of shapes using unions of circles. *Computer Graphics Forum*, 15(3):129–142, 1996.

[21] M. Sharir. On k-sets in arrangements of curves and surfaces. *Discrete Comput. Geom.*, 6:593–613, 1991.