



**HAL**  
open science

# Matching solid shapes in arbitrary dimension via random sampling

Daria Schymura

► **To cite this version:**

Daria Schymura. Matching solid shapes in arbitrary dimension via random sampling. 23rd International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms (AofA'12), 2012, Montreal, Canada. pp.167-178, 10.46298/dmtcs.2992 . hal-01197244

**HAL Id: hal-01197244**

**<https://inria.hal.science/hal-01197244v1>**

Submitted on 11 Sep 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Matching solid shapes in arbitrary dimension via random sampling

Daria Schymura<sup>†</sup>

*Institut für Informatik, Freie Universität Berlin, Germany*

We give simple probabilistic algorithms that approximately maximize the volume of overlap of two solid, i.e. full-dimensional, shapes under translations and rigid motions. The shapes are subsets of  $\mathbb{R}^d$  where  $d \geq 2$ . The algorithms approximate with respect to an pre-specified additive error and succeed with high probability. Apart from measurability assumptions, we only require that points from the shapes can be generated uniformly at random. An important example are shapes given as finite unions of simplices that have pairwise disjoint interiors.

**Keywords:** shape matching, randomized algorithms, computational geometry

## 1 Introduction

We design and analyze simple probabilistic algorithms for matching solid, i.e. full-dimensional, shapes in  $d$ -dimensional Euclidean space ( $d \geq 2$ ), under translations and rigid motions. Given two shapes  $A, B \subset \mathbb{R}^d$ , we search a translation (or a rigid motion)  $t$  such that  $t(A)$  and  $B$  match well. We measure the quality of the match by  $|t(A) \cap B|/|A|$  where  $|\cdot|$  denotes the volume (Lebesgue measure).

We assume that the shapes are given by an oracle that generates points from the shapes uniformly at random (u.a.r.). A common representation of solid shapes, which satisfies this assumption, are finite unions of full-dimensional simplices. A full-dimensional *simplex* is defined as the convex hull of  $d + 1$  affinely independent points in  $\mathbb{R}^d$ . The simplices of a shape are assumed to have pairwise disjoint interiors. Hence, in two dimensions, a shape is a polygonal region, which is given as the union of triangles. It can have holes and is not necessarily connected. This class of shapes is very rich; it contains the homogeneous geometric simplicial complexes as a subclass.

On the input of two shapes  $A$  and  $B$ , the algorithms compute a transformation  $t^*$  such that the volume of overlap of  $t^*(A)$  and  $B$  is approximately maximal. For translations, the idea of the algorithm is as follows. Given two shapes  $A$  and  $B$ , pick a point  $a \in A$  and a point  $b \in B$  u.a.r. This tells us that the translation  $t$  that is given by the vector  $b - a$  maps some part of  $A$  onto some part of  $B$ . We record this as a “vote” for  $t$  and repeat this procedure multiple times, say  $N$  times. Then we determine a densest cluster of the resulting point cloud of translation vectors, and output the center of this cluster. The details of the

<sup>†</sup>Email: [daria.schymura@gmx.net](mailto:daria.schymura@gmx.net) This work was supported by the DFG within the Research Training Group 1408 *Methods for Discrete Structures* and by the DFG Priority Programme 1307 *Algorithm Engineering*.

algorithm are explained in Section 2. The algorithm for matching under rigid motions follows a similar idea. It is explained in Section 3.

We show that the algorithms approximate the maximal volume of overlap under translations (or rigid motions) in the following sense. Let  $t^{\text{opt}}$  be a transformation, i.e. translation or rigid motion, that maximizes the volume of overlap of  $A$  and  $B$ , and let  $t^*$  be a transformation that is computed by our algorithm. Given an error bound  $\varepsilon \in (0, 1)$  and an allowable probability of failure  $p \in (0, 1)$ , we show bounds on the sample size and the clustering size, guaranteeing that  $|t^{\text{opt}}(A) \cap B| - |t^*(A) \cap B| \leq \varepsilon|A|$  with probability at least  $1 - p$ . In this case, we say that  $t^*$  maximizes the volume of overlap of  $A$  and  $B$  up to an additive error of  $\varepsilon|A|$  with probability at least  $1 - p$ . We use  $\varepsilon|A|$  and not just  $\varepsilon$  as error bound because the inequality should be invariant under scaling of both shapes with the same factor.

In the planar case, these algorithms are instances of a probabilistic algorithmic scheme for approximating an optimal match of planar sets. Alt and Scharf [1] analyzed an instance of this algorithmic scheme that compares polygonal curves under translations, rigid motions, similarities, and other subsets of affine transformations.

In a previous publication [2], we considered the 2-dimensional case. Here we not only generalize the results to arbitrary dimension  $d \geq 2$ , but we also give new proofs that improve the bounds on the number of random samples  $N$ . Furthermore we considerably improve the time complexity of the algorithm by showing that a simpler definition of a cluster suffices to guarantee approximation. We represent translations and rigid motions as vectors. Let  $B(t, \delta)$  be the closed ball around the transformation  $t$  of radius  $\delta$  with respect to some chosen metric. The *neighborhood* of a transformation  $t$  is defined to be the ball  $B(t, \delta)$  of some prescribed radius  $\delta$ , which is called the *clustering size*. In [2], that transformation was computed whose neighborhood contained the maximal number of “votes”. This boils down to computing a deepest cell of an arrangement of boxes. For  $N$  boxes in  $\mathbb{R}^k$ , the best known time bound is  $O(N^{k/2}/(\log N)^{k/2-1} \text{polyloglog } N)$  [3]. Here we show that it is sufficient to output the “vote” whose neighborhood contains the maximal number of “votes”, which can be computed by brute force in time  $O(N^2)$  in any dimension. The time bound can be further improved to  $O(N(\log N)^{d-1})$  by using orthogonal range counting queries [5] in the case of translations.

Cheong et al. [4] introduce a general probabilistic framework, which they use for approximating the maximal area of overlap of two unions of  $n$  and  $m$  triangles in the plane, with pre-specified absolute error  $\varepsilon$ , in time  $O(m + (n^2/\varepsilon^4)(\log n)^2)$  for translations and in time  $O(m + (n^3/\varepsilon^8)(\log n)^5)$  for rigid motions. The latter time bound is smaller in their paper, due to a calculation error in the final derivation of the time bound, as was noted in [17]. Their algorithm works with high probability.

For two simple polygons with  $n$  and  $m$  vertices in the plane, Mount et al. [11] show that a translation that maximizes the area of overlap can be computed in time  $O(n^2m^2)$ .

For maximizing the volume of overlap of two unions of simplices under rigid motions, no exact algorithm that runs in polynomial time is known, not even in the plane. Vigneron gives an FPTAS with relative error  $\varepsilon$  for dimension  $d \geq 2$  [17]. For two polyhedra  $P$  and  $Q$  in  $\mathbb{R}^d$ , given as the union of  $m$  and  $n$  simplices, respectively, the algorithm for approximating the maximal volume of overlap has time complexity  $O((\frac{nm}{\varepsilon})^{d^2/2+d/2+1}(\log \frac{nm}{\varepsilon})^{d^2/2+d/2+1})$ . In the plane, this can be improved to time  $O((n^6/\varepsilon^3) \log^4(n/\varepsilon)\beta(n/\varepsilon))$  where  $\beta$  is a very slowly growing function related to the inverse Ackermann function.

We describe the algorithm for matching under translations in Section 2. We also give the main results for translations, which are the approximation property of the algorithm and an upper bound on the time

complexity of the algorithm. In Section 3, we present the algorithm and the main results for rigid motions, which are again the approximation property and an upper bound on the time complexity of the algorithm. Section 4 contains a theorem for approximating Lipschitz continuous density functions uniformly by the empirical measure. We apply this theorem in Section 5 to give the postponed proofs of the approximation properties of both algorithms.

## 2 Matching under translations via random sampling

A translation  $f_t : \mathbb{R}^d \rightarrow \mathbb{R}^d$ ,  $f_t : x \mapsto x + t$  is identified with its translation vector  $t$ , so the translation space equals  $\mathbb{R}^d$ . Observe that an ordered pair of points in  $\mathbb{R}^d$  uniquely determines a translation mapping the first point onto the second. We denote the Euclidean norm by  $|\cdot|$ , and the maximum norm by  $\|\cdot\|_\infty$ . Let  $B_\infty(t, \delta)$  be the closed ball around  $t \in \mathbb{R}^d$  of radius  $\delta$  with respect to  $\|\cdot\|_\infty$ .

We give a pseudocode description of the algorithm for translations, Algorithm 1.

---

### Algorithm 1: ProbMatchT( $A, B, \delta, N$ )

---

**Input:** shapes  $A, B \subset \mathbb{R}^d$ , clustering size  $\delta > 0$ , sample size  $N \in \mathbb{N}$   
 collection  $Q \leftarrow \emptyset$ ;  
**for**  $i = 1 \dots N$  **do**  
     point  $a \leftarrow \text{randomPoint}(A)$ ;   // returns a point from  $A$  u.a.r.  
     point  $b \leftarrow \text{randomPoint}(B)$ ;  
     add( $Q, b - a$ );  
**end**  
**return** FindDensestClusterT( $Q, \delta$ );

---

### Function FindDensestClusterT( $Q, \delta$ )

---

**Input:** collection  $Q$  of points in  $\mathbb{R}^d$ , positive number  $\delta$   
**Output:** point  $t^* \in Q$  such that  $B_\infty(t^*, \delta) \cap Q$  has maximal cardinality

---

Before stating the main result for translations, we introduce some definitions. The boundary  $\partial A$  of a set  $A \subseteq \mathbb{R}^d$  is the set of points that are contained in its closure, but not in its interior. We measure the boundary of  $d$ -dimensional sets by the  $(d - 1)$ -dimensional Hausdorff measure  $\mathcal{H}^{d-1}$ , and denote it slightly sloppily by  $|\partial A|$  most of the time. For a definition of the Hausdorff measure and related definitions, we refer the reader to [8]. The *isoperimetric quotient* of  $A \subset \mathbb{R}^d$  is defined to be  $|\partial A|^d / |A|^{d-1}$ . The isoperimetric quotient can be considered as a certain measure of the *fatness* of a figure  $A$ .

We always assume shapes to be subsets of  $\mathbb{R}^d$  that are Borel sets, have positive, finite Lebesgue measure and have a  $\mathcal{H}^{d-1}$ -measurable and  $(\mathcal{H}^{d-1}, d - 1)$ -rectifiable boundary.

**Theorem 1 (Approximation property of Algorithm 1)** *Let  $A, B \subset \mathbb{R}^d$  be shapes in constant dimension  $d$ , and let  $\varepsilon, p \in (0, 1)$  be parameters. There exists a universal constant  $C > 0$  such that, if  $0 < \delta \leq \frac{2\sqrt{d}}{3(d+1)} \varepsilon \frac{|A|}{|\partial A|}$  and  $N \geq C\varepsilon^{-2} \delta^{-2d} |B|^2 \log \frac{2}{p}$ , then Algorithm 1 with input  $(A, B, \delta, N)$  computes a translation that maximizes the volume of overlap of  $A$  and  $B$  up to an additive error of  $\varepsilon|A|$  with probability at least  $1 - p$ .*

The universal constant  $C$  can be deduced from the proofs. We postpone the proof of this theorem to Section 5. The algorithm can also provide a good estimate of the maximal volume of overlap. Let  $t^*$  be the output of the algorithm, i.e. the random translation maximizing the number of points in  $B_\infty(t, \delta) \cap Q$ . The number  $\#(B_\infty(t^*, \delta) \cap Q) \cdot |A| \cdot |B| (2\delta)^{-d} N^{-1}$  differs from  $\max_{t \in \mathbb{R}^d} |t(A) \cap B|$  by at most  $\frac{3}{2}\varepsilon|A|$ , with probability at least  $1 - p$ .

Our algorithm can be applied to all shapes from which we can generate random sample points from the uniform distribution. For unions of simplices, the runtime of our method depends only linearly on the number of vertices, but it depends more significantly on fatness parameters as the isoperimetric quotient of one of the shapes and, in the case of rigid motions, on the ratio  $\text{diam}(A)^d/|A|$ .

**Theorem 2 (Runtime for translations)** *Let  $A$  and  $B$  be shapes in constant dimension  $d$ , and let  $\varepsilon, p \in (0, 1)$  be parameters. Assume that we are given an upper bound  $K_A$  on the isoperimetric quotient of  $A$ , a lower bound  $m_A$  on  $|A|$  and an upper bound  $M_B$  on  $|B|$ . Assume further that  $N$  points from a shape can be generated u.a.r. in time  $T(N)$ .*

*Then a translation that maximizes the volume of overlap of the shapes  $A$  and  $B$  up to an additive error of  $\varepsilon|A|$  with probability  $\geq 1 - p$  can be computed in time  $O(T(N) + N(\log N)^{d-1})$  where  $N = O(\varepsilon^{-(2d+2)} (K_A M_B / m_A)^2 \log \frac{2}{p})$ .*

*If  $A$  and  $B$  are finite unions of at most  $n$  simplices that have pairwise disjoint interiors, then  $T(N) = O(n + N)$  and the bounds  $K_A, m_A$  and  $M_B$  can be computed in time  $O(n)$ .*

**Proof of Theorem 2:** By assumption, we are given an upper bound  $K_A$  on the isoperimetric quotient of  $A$ , a lower bound  $m_A$  on  $|A|$ , and an upper bound  $M_B$  on  $|B|$ . Hence we can compute a clustering size  $\delta$  and a sample size  $N$  that fulfill the bounds in Theorem 1 in constant time:

$$\delta := \frac{2\sqrt{d}}{3(d+1)} \varepsilon (m_A/K_A)^{1/d} \leq \frac{2\sqrt{d}}{3(d+1)} \varepsilon \frac{|A|}{|\partial A|},$$

$$N := \left\lceil C \varepsilon^{-2} \delta^{-2d} M_B^2 \log \frac{2}{p} \right\rceil \geq C \varepsilon^{-2} \delta^{-2d} |B|^2 \log \frac{2}{p}.$$

With input  $(A, B, \delta, N)$ , Algorithm 1 computes a translation with the desired property, so we have to upper bound the runtime of Algorithm 1. By assumption, we can generate  $N$  points from a shape u.a.r. in time  $T(N)$ . For adding  $N$  points to a collection  $Q$  and then computing  $\text{FindDensestClusterT}(Q, \delta)$ , we do the following. Let  $Q = \{t_1, \dots, t_N\}$ . We build a standard orthogonal range counting query data structure for  $t_1, \dots, t_N$  in time  $O(N(\log N)^{d-1})$ . We query the data structure  $N$  times with the cubes  $B_\infty(t_i, \delta)$ . Each query takes time  $O((\log N)^{d-1})$ . Hence a translation  $t_j$  maximizing  $\#(B_\infty(t_i, \delta) \cap Q)$  can be computed in time  $O(N(\log N)^{d-1})$  for all  $t_1, \dots, t_N \in \mathbb{R}^d$  and  $\delta > 0$ . For details and references on range counting queries, see for example [5].

It remains to show that  $T(N) = O(n + N)$  and the bounds  $K_A, m_A$  and  $M_B$  can be computed in time  $O(n)$  in the case of unions of simplices. Since we regard  $d$  as a constant, the volume of a simplex can be computed in constant time, and we can compute  $m_A := |A|$  and  $M_B := |B|$  in time linear in  $n$ . We do not know how to compute  $|\partial A|$  in time linear in  $n$  since  $A$  is given as a set of simplices. But we can compute an upper bound on  $|\partial A|$  in linear time by adding all  $(d-1)$ -dimensional volumes of the facets of the simplices. Obviously, this bound can be bad as many of the facets may lie in the interior of the shape, but it can be sharp as it is matched for collections of pairwise disjoint simplices.

To generate points from  $A$  u.a.r., first draw a simplex from  $A$  with probability proportional to the volume of the simplex. Then we draw a point from this simplex u.a.r. For drawing a simplex, we use the alias

method [18]. Let  $A$  be the union of the simplices  $\Delta_1, \dots, \Delta_n$  that have pairwise disjoint interiors. Let  $p_i := \frac{|\Delta_i|}{|A|}$ . The vector  $(p_1, \dots, p_n)$  describes the discrete probability distribution according to which we have to pick a random point from simplex  $\Delta_i$ . The alias method can be implemented such that after  $O(n)$  preprocessing time, we can sample random points from  $(p_1, \dots, p_n)$  in time  $O(1)$  [14]. Generating a point from a  $d$ -dimensional simplex u.a.r. can be done in  $O(d \log d)$  time by generating  $d$  numbers in  $[0, 1]$  u.a.r., sorting them  $0 \leq x_1 < \dots < x_d \leq 1$  and taking the spacings  $x_1, x_2 - x_1, \dots, x_d - x_{d-1}, 1 - x_d$  as barycentric coordinates of the point in the simplex [6]. Thus generating  $N$  random points in  $A$  and  $B$  takes  $T(N) = O(n + N)$  time.  $\square$

### 3 Matching under rigid motions via random sampling

A *rigid motion*  $r : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is a map given by  $r(x) = Mx + t$  for some rotation matrix  $M \in \mathbb{R}^{d \times d}$  and some translation vector  $t \in \mathbb{R}^d$ . A matrix  $M$  is contained in the group of rotation matrices  $SO(d) \subset \mathbb{R}^{d \times d}$  if it is orthogonal, meaning  $M^T = M^{-1}$ , and  $\det M = 1$ . We identify each rigid motion with the pair of its rotation matrix and translation vector. Thus the space of rigid motions equals  $SO(d) \times \mathbb{R}^d$ .

The algorithm for rigid motions works similarly as the algorithm for translations. We draw a rotation matrix  $M \in SO(d)$ , a point  $a \in A$  and a point  $b \in B$  u.a.r. Then we register the unique rigid motion that has  $M$  as rotation matrix and maps  $a$  onto  $b$  as a “vote” in the transformation space. After many rounds, we determine the best cluster in the space of rigid motions.

Within the algorithm, rotation matrices are drawn u.a.r. To define the uniform distribution on  $SO(d)$ , a volume has to be defined. The group  $SO(d)$  is a  $\binom{d}{2}$ -dimensional manifold. We measure the volume  $|\cdot|$  in  $SO(d)$  by the  $\binom{d}{2}$ -dimensional Hausdorff measure [8], which is a Haar measure on  $SO(d)$ .

For a matrix  $M = (m_{ij})_{1 \leq i, j \leq d}$ , let  $\|M\|_2 = \sqrt{\sum_{1 \leq i, j \leq d} (m_{ij})^2}$  be the Frobenius norm. For  $M \in SO(d)$  and  $t \in \mathbb{R}^d$ , let  $B_2(M, \delta)$  and  $B_2(t, \delta)$  be the closed balls of radius  $\delta$  with respect to the metrics induced by the Frobenius and the Euclidean norm. We define a metric on the space of rigid motions, which depends on the input shape  $A$ , by

$$\text{dist}((M, p), (N, q)) = \max\{\Delta_A \|M - N\|_2, |p - q|\}$$

for some given  $\Delta_A \geq \text{diam}(A)$ . We define a  $\delta$ -neighborhood of a rigid motion  $(M, t)$  by  $B((M, t), \delta) = B_2(M, \delta/\Delta_A) \times B_2(t, \delta)$ , which is the closed ball with respect to the metric  $\text{dist}$  around  $(M, t)$  of radius  $\delta$ . The radius of the rotational part of the neighborhood depends on the diameter of  $A$  because it should not change if  $A$  is scaled. The “rotational distance” of shapes does not depend on their absolute size. The center of a best cluster is a random rigid motion (“vote”) whose neighborhood contains the maximal number of random rigid motions. We give a pseudocode description of the algorithm for rigid motions, Algorithm 3.

**Theorem 3 (Approximation property of Algorithm 3)** *Let  $A, B \subset \mathbb{R}^d$  be shapes in constant dimension  $d$ , and let  $\varepsilon, p \in (0, 1)$  be parameters. There are constants  $C, C' > 0$  such that, if  $0 < \delta \leq C\varepsilon \frac{|A|}{|\partial A|}$  and  $\Delta \geq \text{diam}(A)$  and  $N \geq C'\varepsilon^{-2} \delta^{-d^2-d} \Delta^{d^2-d} |B|^2 \log \frac{2}{p}$ , then Algorithm 3 with input  $(A, B, \delta, N, \Delta)$  computes a rigid motion that maximizes the volume of overlap of  $A$  and  $B$  up to an additive error of  $\varepsilon|A|$  with probability  $\geq 1 - p$ .*

**Algorithm 3:** ProbMatchRM( $A, B, \delta, N, \Delta$ )

---

**Input:** shapes  $A, B \subset \mathbb{R}^d$ , clustering size  $\delta > 0$ , sample size  $N \in \mathbb{N}$ , diameter bound  $\Delta$   
collection  $Q \leftarrow \emptyset$ ;  
**for**  $i = 1 \dots N$  **do**  
    // randomRotation() returns a rotation matrix u.a.r.  
    rotation matrix  $M \leftarrow \text{randomRotation}()$ ;  
    point  $a \leftarrow \text{randomPoint}(A)$ ;  
    point  $b \leftarrow \text{randomPoint}(B)$ ;  
    add( $Q, (M, b - Ma)$ );  
**end**  
**return** FindDensestClusterRM( $Q, \delta, \Delta$ );

---

**Function** FindDensestClusterRM( $Q, \delta, \Delta$ )

---

**Input:** collection  $Q$  of points in  $\mathbb{R}^{d \times d} \times \mathbb{R}^d$ , positive numbers  $\delta$  and  $\Delta$   
**Output:** point  $(M, t) \in Q$  such that the neighborhood  $B_2(M, \delta/\Delta) \times B_2(t, \delta)$  of  $(M, t)$  contains a maximal number of points from  $Q$

---

We postpone the proof of this theorem to Section 5 and continue by discussing an upper bound on the runtime of Algorithm 3.

**Theorem 4 (Runtime for rigid motions)** *Let  $A$  and  $B$  be shapes in constant dimension  $d$ , and let  $\varepsilon, p \in (0, 1)$  be parameters. Assume that we are given an upper bound  $\Delta_A$  on  $\text{diam}(A)$ , an upper bound  $K_A$  on the isoperimetric quotient of  $A$ , a lower bound  $m_A$  on  $|A|$  and an upper bound  $M_B$  on  $|B|$ . Assume further that  $N$  points from a shape can be generated u.a.r. in time  $T(N)$ . Then a rigid motion that maximizes the volume of overlap of  $A$  and  $B$  up to an additive error of  $\varepsilon|A|$  with probability  $\geq 1 - p$  can be computed in time  $O(T(N) + N^2)$  where  $N = O(\varepsilon^{-(d^2+d+2)} K_A^{d+1} D_A^{d-1} (M_B/m_A)^2 \log \frac{2}{p})$  and  $D_A := \Delta_A^d/m_A$  is an upper bound on the ratio  $\text{diam}(A)^d/|A|$ .*

Note that a bound  $\Delta_A$  can be computed in linear time in the case of unions of simplices by computing the maximal distance  $D$  of a vertex from the origin. W.l.o.g. the shape contains the origin. Then  $\frac{1}{2} \text{diam}(A) \leq D \leq \text{diam}(A)$ , and we take  $\Delta_A = 2D$  as the upper bound on the diameter of  $A$ . Hence the bounds  $\Delta_A, K_A, m_A$  and  $m_B$  can be computed in linear time.

**Proof of Theorem 4:** We can compute in constant time a clustering size  $\delta$  and a sample size  $N$  that guarantee approximation by Theorem 3, using the given bounds  $\Delta_A, K_A, m_A$  and  $M_B$ , similarly as in the case of translations.

There are many methods to generate uniformly distributed random orthogonal matrices described in the literature, the first being [10, 15]. A simple method to generate an orthogonal matrix u.a.r. is to compute  $\binom{d}{2}$  random rotations of two axes, while all other axes are fixed [6]. The time needed is constant since we regard the dimension as constant.

To determine the densest cluster, we simply check by brute force for every pair  $\{r, r'\}$  of random rigid motions that is generated in the algorithm if  $\text{dist}(r, r') \leq \delta$ . One test can be done in constant time. So we can determine a densest cluster in time  $O(N^2)$ .  $\square$

In the 2-dimensional case, the results can be significantly improved by representing a rigid motion by the pair of its rotation angle (instead of the rotation matrix) and the translation vector, as it was done in [2]. The running time then becomes  $O(T(N) + N(\log N)^2)$ . In dimension 3, the running time can be improved to  $O(T(N) + N(\log N)^5)$  by representing the rotation by three angles. In both cases, the improvement comes from the fact that the different representation of rigid motions allows to use neighborhoods for the clustering that are boxes in  $\mathbb{R}^3$  or  $\mathbb{R}^6$ , respectively. Then orthogonal range counting queries can be used as in the case of translations. For details, see [13, Section 8.5].

## 4 The probabilistic toolbox

Let  $\mu$  be the probability distribution on the transformation space that is induced by the random experiment in the algorithm, and let  $\mu_N$  be the empirical measure. The main idea is to prove that the density function of  $\mu$  is proportional to the objective function, that is the function that maps a transformation  $t$  to  $|t(A) \cap B|$ . Thus the goal is to find a transformation at which the density function is approximately maximal.

Conceptually, the density function is approximated in a two step process. Let  $B(t, \delta)$  be a ball of radius  $\delta$ , centered at  $t$ , with respect to some metric. For translations, we choose the metric that is induced by the maximum norm. First,  $f(t) \cdot |B(t, \delta)|$  is close to  $\mu(B(t, \delta))$  if  $f$  is nice enough and  $\delta$  is sufficiently small. Second, the probability of a small cube  $\mu(B(t, \delta))$  is close to  $\mu_N(B(t, \delta))$  if  $N$  is sufficiently large.

The analysis of the algorithm is based on these simple ideas. The details are hidden in the following longish theorem that follows from theorems in [7, Chapters 2 and 4]. For the definition of the Vapnik-Chervonenkis dimension (VC dimension), see [16].

**Theorem 5 (Probabilistic toolbox)** *Let  $\Omega \subseteq \mathbb{R}^k$  be a metric space, let  $\delta > 0$ , and let  $\mathcal{B}_\delta$  be the set of balls of radius  $\delta$  in  $\Omega$ . Let  $\text{vol}$  be a measure on  $\Omega$  such that there is  $v_\delta > 0$  with  $\text{vol}(B) = v_\delta$  for all  $B \in \mathcal{B}_\delta$ . Assume further that  $\mathcal{B}$  has finite VC dimension  $V$ .*

*Let  $\mu$  be a probability measure on  $\Omega$  that has a Lipschitz continuous density function  $f$  with Lipschitz constant  $L$ . Let  $X_1, \dots, X_N$  be i.i.d. random variables taking values in  $\Omega$  with common distribution  $\mu$  and empirical measure  $\mu_N$ .*

*Let  $j \in \{1, \dots, N\}$  be such that  $\mu_N(B(X_j, \delta)) = \max\{\mu_N(B(X_i, \delta)) : 1 \leq i \leq N\}$ . There is a universal constant  $c$  such that, for all  $\tau > 0$ , with probability  $\geq 1 - 2e^{-2N\tau^2}$ , we have*

$$f(X_j) \geq f(x) - 2(c\sqrt{V/N} + \tau)/v_\delta - 3L\delta \quad \text{for all } x \in \Omega. \quad (1)$$

We cite two results from [7], which we will use to prove Theorem 5.

**Theorem 6 ([7, Chapter 2])** *Let  $N \in \mathbb{N}$  and let  $X_1, \dots, X_N$  be i.i.d. random variables taking values in  $\mathbb{R}^k$  with common distribution  $\mu$  and empirical distribution  $\mu_N$ . Let  $\mathcal{C}$  be a class of subsets of  $\mathbb{R}^k$ . For all  $\tau > 0$ ,*

$$P\left(\left|\sup_{C \in \mathcal{C}} |\mu_N(C) - \mu(C)| - \mathbf{E}\left(\sup_{C \in \mathcal{C}} |\mu_N(C) - \mu(C)|\right)\right| > \tau\right) \leq 2e^{-2N\tau^2}.$$

**Theorem 7 ([7, Chapter 4])** *Let  $X_1, \dots, X_N$  be i.i.d. random variables taking values in  $\mathbb{R}^k$  with common distribution  $\mu$  and empirical distribution  $\mu_N$ . Let  $\mathcal{C}$  be a class of subsets of  $\mathbb{R}^k$  with VC dimension  $V$ . There is a universal constant  $c$  such that*

$$\mathbf{E}\left(\sup_{C \in \mathcal{C}} |\mu_N(C) - \mu(C)|\right) \leq c\sqrt{V/N}.$$



The constant  $c$  could be bounded explicitly. Using an inequality of Haussler [9] (cited from [7, Chapter 4]), one can show  $c < 75$ .

**Proof of Theorem 5.:** Because of the Lipschitz continuity of  $f$  and the definition of the density function, for all  $B \in \mathcal{B}_\delta$  with center  $x$  holds

$$|f(x) - \mu(B)/v_\delta| \leq L\delta. \quad (2)$$

By the triangle inequality and Theorems 6 and 7, we know that with probability at least  $1 - 2e^{-2N\tau^2}$  for all  $B \in \mathcal{B}_\delta$

$$|\mu(B) - \mu_N(B)| \leq c\sqrt{V/N} + \tau. \quad (3)$$

Let  $\eta = (c\sqrt{V/N} + \tau)/v_\delta$ . If  $f(x) \leq \eta + L\delta$  for all  $x \in \Omega$ , then Equation (1) is true because  $f$  is non-negative. Assume that there is an  $x \in \Omega$  such that  $f(x) > \eta + L\delta$ . Let  $\bar{x} \in \Omega$  be a point for which  $f$  is maximal. If  $f$  does not attain its supremum, then let  $\varepsilon' > 0$  and let  $\bar{x}$  be a point such that  $f(\bar{x}) > \eta + L\delta$  and  $f(\bar{x}) + \varepsilon' > f(x)$  for all  $x \in \Omega$ .

Let  $\bar{B} = B(\bar{x}, \delta)$ . By the triangle inequality and Inequalities (2) and (3), we have

$$f(\bar{x}) \leq \mu_N(\bar{B})/v_\delta + \eta + L\delta,$$

which implies  $\mu_N(\bar{B}) > 0$ . Therefore there is an  $i \in \{1, \dots, N\}$  such that  $X_i \in \bar{B}$ . Since  $f$  is Lipschitz continuous, we have

$$|f(\bar{x}) - f(X_i)| \leq L\delta. \quad (4)$$

We put things together now. First we use Inequality (4). Second we use Inequalities (2), (3) and the triangle inequality. Third we use the assumption on  $X_j$ . Lastly we use Inequalities (2), (3) and the triangle inequality again. We get that with probability at least  $1 - 2e^{-2N\tau^2}$  the following holds:

$$\begin{aligned} f(\bar{x}) &\leq f(X_i) + L\delta \\ &\leq \mu_N(B(X_i, \delta))/v_\delta + \eta + 2L\delta \\ &\leq \mu_N(B(X_j, \delta))/v_\delta + \eta + 2L\delta \\ &\leq f(X_j) + 2\eta + 3L\delta. \end{aligned}$$

Since this holds for all small  $\varepsilon' > 0$ , it implies  $f(x) \leq f(X_j) + 2\eta + 3L\delta$  for all  $x \in \Omega$  and thus Inequality (1).  $\square$

## 5 Analysis of the algorithms (Proof of Theorems 1 and 3)

The key lemma for applying Theorem 5 to the analysis of Algorithms 1 and 3 states that the density function of  $\mu$  is proportional to the objective function.

### Lemma 8 (Key lemma)

- (i) Let  $X$  be the random vector on  $\mathbb{R}^d$  that draws translations  $t = b - a$  where  $(a, b) \in A \times B \subset \mathbb{R}^{2d}$  is drawn u.a.r. The density function of  $X$  is given by  $f(t) = \frac{|t(A) \cap B|}{|A| \cdot |B|}$ .

(ii) Let  $Y$  be the random vector that draws rigid motions  $(M, b - Ma) \in SO(d) \times \mathbb{R}^d$  where  $(M, a, b) \in SO(d) \times A \times B$  is drawn u.a.r. The density function of  $Y$  is given by  $g(r) = \frac{|r(A) \cap B|}{|SO(d)| \cdot |A| \cdot |B|}$ .

**Proof:** (i) *Translations.* Let  $Z$  be the random vector that draws points  $(a, b) \in \mathbb{R}^d \times \mathbb{R}^d$  u.a.r. Its density function  $f_Z$  is given by  $f_Z(x) = \frac{\chi_{A \times B}(x)}{|A| |B|}$ . Let  $\varphi : A \times B \rightarrow A \times (-A \oplus B)$  be the function defined by  $\varphi(a, b) = (a, b - a)$ . Let  $J = \begin{pmatrix} I_d & 0 \\ -I_d & I_d \end{pmatrix}$  where  $I_d$  denotes the  $(d \times d)$ -identity matrix. We have  $\varphi(x) = Jx$  and  $\det J = 1$ . By a simple transformation rule for integrals, the density function of  $Y = \varphi(Z)$  equals  $f_Y(a, t) = \frac{\chi_{A \times B}(a, a+t)}{|A| |B|}$ . Let  $\pi$  be the projection defined by  $\pi : A \times (-A \oplus B) \rightarrow (-A \oplus B)$ ,  $\pi : (a, t) \mapsto t$ , and define  $X = \pi(Y)$ . The density function  $f$  of  $X$  equals  $\int_{\mathbb{R}^d} f_Y(a, t) da$ . Since  $\chi_{A \times B}(a, a+t) = \chi_A(a) \chi_{B-t}(a) = \chi_{A \cap (B-t)}(a)$ , we have

$$f(t) = \int_{\mathbb{R}^d} \frac{\chi_{A \cap (B-t)}(a)}{|A| |B|} da = \frac{|A \cap (B-t)|}{|A| |B|} = \frac{|(A+t) \cap B|}{|A| |B|}.$$

(ii) *Rigid motions.* The random experiment consists in drawing points from  $\Omega = SO(d) \times A \times B$  uniformly at random. Our goal is to determine the density function  $f_Y$  of the random variable

$$Y : \Omega \rightarrow SO(d) \times \mathbb{R}^d, \quad Y : (M, a, b) \mapsto (M, b - Ma).$$

The density function of  $Y$  can be expressed in terms of the conditional densities of the two random variables

$$\begin{aligned} Y_{SO(d)} : \Omega &\rightarrow SO(d), & Y_{SO(d)} : (M, a, b) &\mapsto M, \\ Y_{A \times B} : \Omega &\rightarrow \mathbb{R}^d, & Y_{A \times B} : (M, a, b) &\mapsto b - Ma. \end{aligned}$$

The density function of  $Y$  is the joint density of the random variables  $Y_{SO(d)}$  and  $Y_{A \times B}$ . The conditional density function can also be expressed in terms of the joint density function

$$f_{A \times B}(t \mid Y_{SO(d)} = M) = f_Y(M, t) / f_{SO(d)}(M). \quad (5)$$

Clearly,  $f_{SO(d)} \equiv 1/|SO(d)|$ . The conditional density function  $f_{A \times B}(t \mid Y_{SO(d)} = M)$  equals the density function of the translational case for the shapes  $MA$  and  $B$  and thus is  $\frac{|(MA+t) \cap B|}{|A| |B|}$ . Therefore,  $f_Y(M, t) = \frac{|(MA+t) \cap B|}{|A| |B| |SO(d)|}$ .  $\square$

Furthermore we have to show that the density functions  $f$  and  $g$  are Lipschitz continuous. We show this by using the following theorem that gives upper bounds on the volume of sets  $A \triangle t(A)$  for translations and rotations  $t$ . Therein  $\triangle$  denotes the symmetric difference. Let  $\mathcal{H}^d$  be the  $d$ -dimensional Hausdorff measure. For Lebesgue measurable sets in  $\mathbb{R}^d$ , the  $d$ -dimensional Hausdorff measure and the Lebesgue measure coincide.

**Theorem 9** [12] *Let  $A \subset \mathbb{R}^d$  be bounded.*

- (i) *For all translation vectors  $t \in \mathbb{R}^d$ , we have  $\mathcal{H}^d(A \triangle (A+t)) \leq |t| \mathcal{H}^{d-1}(\partial A)$ .*
- (ii) *Assume that  $A$  has an  $(\mathcal{H}^{d-1}, d-1)$ -rectifiable boundary. For all rotation matrices  $M \in \mathbb{R}^{d \times d}$ , we have  $\mathcal{H}^d(A \triangle MA) \leq \max_{a \in \partial A} |a - Ma| \mathcal{H}^{d-1}(\partial A)$ .*

**Corollary 10**

- (i) The function  $f$ , as defined in Lemma 8, is Lipschitz continuous with constant  $L = \frac{\sqrt{d}|\partial A|}{2|A||B|}$  with respect to the metric that is induced by the maximum norm.
- (ii) The function  $g$ , as defined in Lemma 8, is Lipschitz continuous with constant  $L = \frac{|\partial A|}{|SO(d)||A||B|}$  with respect to the metric defined by  $\text{dist}((M, p), (N, q)) = \max\{\Delta_A \|M - N\|_2, |p - q|\}$ .

**Proof:** Let  $s, t : \mathbb{R}^d \rightarrow \mathbb{R}^d$  be translations or rigid motions. Then

$$\begin{aligned} ||s(A) \cap B| - |t(A) \cap B|| &= |(s(A) \setminus t(A)) \cap B| - |(t(A) \setminus s(A)) \cap B| \\ &\leq \max\{|t(A) \setminus s(A)|, |s(A) \setminus t(A)|\} \\ &= \frac{1}{2}|s(A) \Delta t(A)| = \frac{1}{2}|A \Delta (s^{-1} \circ t)(A)|. \end{aligned}$$

(i) *Translations.* The map  $s^{-1} \circ t$  corresponds to the translation vector  $t - s$ . By the above inequality and Theorem 9, we have  $||s(A) \cap B| - |t(A) \cap B|| \leq \frac{1}{2}|t - s| \cdot |\partial A|$ . Because of  $|t - s| \leq \sqrt{d}\|t - s\|_\infty$ , the claim is proved.

(ii) *Rigid motions.* Let  $s = (M, p)$  and  $t = (N, q)$ . The map  $s^{-1} \circ t$  is a rigid motion with rotation matrix  $N^{-1}M$  and translation vector  $x := N^{-1}(p - q)$ . We have  $|x| = |p - q|$  and  $|a - N^{-1}Ma| = |Na - Ma|$ . Therefore, by the above inequality, the triangle inequality and Theorem 9, we have

$$\begin{aligned} ||s(A) \cap B| - |t(A) \cap B|| &\leq \frac{1}{2}|A \Delta (N^{-1}MA + x)| \\ &\leq \frac{1}{2}(|A \Delta (N^{-1}MA)| + |(N^{-1}MA) \Delta (N^{-1}MA + x)|) \\ &\leq \frac{1}{2}(|p - q| + \max_{a \in \partial A} |Na - Ma|) |\partial A|. \\ &\leq \frac{1}{2}(|p - q| + \max_{a \in \partial A} \|N - M\|_2 |a|) |\partial A|. \end{aligned}$$

Since w.l.o.g.  $A$  contains the origin, we have  $\max_{a \in \partial A} |a| \leq \text{diam}(A)$ . This implies

$$|p - q| + \max_{a \in \partial A} \|N - M\|_2 |a| \leq 2 \text{dist}(s, t),$$

and thus the claim.  $\square$

**Proof of Theorem 1:** We apply the probabilistic toolbox Theorem 5 to the translation space  $(\mathbb{R}^d, \|\cdot\|_\infty)$ . Let  $\mathcal{B}_\delta$  be the set of closed balls that have radius  $\delta$ . For every  $B \in \mathcal{B}_\delta$ , we have  $|B| = 2^d \delta^d$ . The VC dimension of the class of rectangles in  $\mathbb{R}^d$  equals  $2d$  [7].

Let  $X_1, X_2, \dots, X_N$  be the random translations in the algorithm. Denote their common distribution by  $\mu$ . By Lemma 8, the measure  $\mu$  has a density function  $f$  that is given by  $f(t) = \frac{|t(A) \cap B|}{|A||B|}$ , and by Lemma 10,  $f$  is Lipschitz continuous with constant  $L = \frac{\sqrt{d}|\partial A|}{2|A||B|}$ .

Let  $\mu_N$  be the empirical measure of  $\mu$ . Algorithm 1 computes a  $B^* \in \mathcal{B}_\delta$  such that  $\mu_N(B^*) = \max\{\mu_N(B(X_i, \delta)) : i = 1, \dots, N\}$ . Let  $t^*$  be the center of  $B^*$ . By Theorem 5, there is a constant  $c > 0$  such that, for all  $\tau > 0$ , with probability at least  $1 - 2e^{-2N\tau^2}$ , we have

$$f(t^*) \geq f(t) - \left( \frac{2c\sqrt{2d}}{\sqrt{N}(2\delta)^d} + \frac{2\tau}{(2\delta)^d} + 3L\delta \right) \quad \text{for all } t \in \mathbb{R}^d. \quad (6)$$

Let  $\varepsilon, p \in (0, 1)$ . We want to keep  $N$  as small as possible as the runtime of our algorithm grows with  $N$ . We determine the minimal value of  $N$  and a compatible value of  $\delta$  for which we can guarantee that with probability at least  $1 - p$  the error is at most  $\varepsilon|A|$ . For the sake of clarity, we compute  $N$  only up to a constant.

Since  $p \in (0, 1)$  is the allowable probability of failure, we should have  $2e^{-2N\tau^2} \leq p$ , which is equivalent to  $\tau \geq \sqrt{\frac{1}{2N} \log \frac{2}{p}}$ . The larger  $\tau$  is, the larger is the error term in Inequality (6), so we set  $\tau = \sqrt{\frac{1}{2N} \log \frac{2}{p}}$ . Substituting  $\tau, L$  and the definition of  $f$  shows

$$|t^*(A) \cap B| \geq |t(A) \cap B| - \left( \frac{\sqrt{2}|A||B|}{\sqrt{N}(2\delta)^d} \left( 2c\sqrt{d} + \sqrt{\log \frac{2}{p}} \right) + \frac{3}{2}\sqrt{d}\delta|\partial A| \right). \quad (7)$$

We want the additive error in Inequality (7) to be at most  $\varepsilon|A|$ . Therefore it is necessary that  $\delta < \frac{2\varepsilon|A|}{3\sqrt{d}|\partial A|}$ . Let  $\eta \in (0, 1)$  be such that  $\delta = \eta \frac{2\varepsilon|A|}{3\sqrt{d}|\partial A|}$ . Then as condition for  $N$  we get the following:

$$\frac{\sqrt{2}|A||B|}{\sqrt{N}(2\delta)^d} \left( 2c\sqrt{d} + \sqrt{\log \frac{2}{p}} \right) < (1 - \eta)\varepsilon|A|.$$

There is a universal constant  $C > 0$ , such that this condition is fulfilled if

$$N \geq C(1 - \eta)^{-2} \varepsilon^{-2} \delta^{-2d} |B|^2 \log \frac{2}{p}.$$

Differentiating with respect to  $\eta$  shows that  $\eta = \frac{d}{d+1}$  minimizes the bound on  $N$ , so we use this value, which gives  $\delta \leq \frac{2\sqrt{d}}{3(d+1)} \frac{\varepsilon|A|}{|\partial A|}$ .  $\square$

**Proof of Theorem 3:** The proof works similarly as in the case of translations. We apply Theorem 5 to  $(SO(d) \times \mathbb{R}^d, \text{dist})$ , using Lemma 8 and Corollary 10. The class  $\mathcal{C} = \{B(r, \delta) : \delta > 0, r \in SO(d) \times \mathbb{R}^d\}$  has VC dimension  $\leq 3(d^2 + d + 2) \log(3(d^2 + d + 2))$  [13, Lemma 8.11]. For fixed  $\delta$ , the balls  $B(r, \delta)$  have the same volume for all  $r$  since we measure  $SO(d)$  by a Haar measure. For sufficiently small  $\delta$ , the volume of  $B(r, \delta)$  is bounded from below by  $C\Delta_A^{-\binom{d}{2}} \delta^{\binom{d+1}{2}}$  for some  $C$  depending on  $d$  [13, Appendix A]. Due to lack of space, we omit the calculations. They can be found in [13, Chapter 8].  $\square$

## 6 Open problem

We studied a probabilistic algorithmic scheme for the case of matching solid shapes in arbitrary dimension under translations and rigid motions. A similarity  $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is a map given by  $\sigma(x) = \lambda Mx + t$  where  $\lambda \in \mathbb{R}$ ,  $M \in SO(d)$  and  $t \in \mathbb{R}^d$ . It is natural to apply the algorithmic scheme to similarities since an ordered pair of points in each shape  $(a_1, a_2) \in A^2$ ,  $(b_1, b_2) \in B^2$  uniquely determines a similarity that maps  $a_1$  onto  $b_1$  and  $a_2$  onto  $b_2$ . It is an open problem which objective function the resulting algorithm optimizes.

## References

- [1] H. Alt and L. Scharf. Shape matching by random sampling. In *WALCOM: Algorithms and computation*, volume 5431 of *Lecture Notes in Comput. Sci.*, pages 381–393. 2009.
- [2] H. Alt, L. Scharf, and D. Schymura. Probabilistic matching of planar regions. *Comput. Geom.*, 43:99–114, 2010.
- [3] T. Chan. A (slightly) faster algorithm for Klee’s measure problem. *Comput. Geom.*, 43(3):243 – 250, 2010.
- [4] O. Cheong, A. Efrat, and S. Har-Peled. Finding a guard that sees most and a shop that sells most. *Discrete Comput. Geom.*, 37:545–563, 2007.
- [5] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry*. Springer, third edition, 2008.
- [6] L. Devroye. *Non-Uniform Random Variate Generation*. Springer, 1986.
- [7] L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer, 2001.
- [8] H. Federer. *Geometric Measure Theory*. Springer, 1969.
- [9] D. Haussler. Sphere packing numbers for subsets of the Boolean  $n$ -cube with bounded Vapnik-Chervonenkis dimension. *J. Combin. Theory Ser. A*, 69(2):217–232, 1995.
- [10] R. Heiberger. Generation of random orthogonal matrices. *J. R. Stat. Soc. Ser. C.*, 27:199–206, 1978.
- [11] D. Mount, R. Silverman, and A. Wu. On the area of overlap of translated polygons. *Computer Vision and Image Understanding*, 64:53–61, 1996.
- [12] D. Schymura. An upper bound on the volume of the symmetric difference of a body and a congruent copy. Submitted. Preprint at <http://arxiv.org/abs/1010.2446>, 2010.
- [13] D. Schymura. *Probabilistic matching of solid shapes in arbitrary dimension*. PhD thesis, Freie Universität Berlin, Institut für Informatik, November 2011. [http://www.diss.fu-berlin.de/diss/receive/FUDISS\\_thesis\\_000000034542](http://www.diss.fu-berlin.de/diss/receive/FUDISS_thesis_000000034542).
- [14] W. Smith. How to sample from a probability distribution, 2002. <http://introcs.cs.princeton.edu/java/98simulation/smith-sampling.ps>.
- [15] M. Tanner and R. Thisted. A remark on AS127. Generation of random orthogonal matrices. *J. R. Stat. Soc. Ser. C. Appl. Stat.*, 31:190–192, 1982.
- [16] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theor. Probab. Appl.*, 16:264–280, 1971.
- [17] A. Vigneron. Geometric optimization and sums of algebraic functions. In *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, pages 906–917, 2010.
- [18] A. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Softw.*, 3:253–256, 1977.