Shape Matching With Reference Points

Dissertation zur Erlangung des Doktorgrades

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

2007

von

Oliver Klein

Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
oklein@inf.fu-berlin.de

gefördert durch die DFG im Rahmen des Europäischen Graduiertenkollegs
Combinatorics, Geometry, and Computation (GRK 588)

Betreuer: Prof. Dr. Günter Rote
Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
Germany
rote@inf.fu-berlin.de

Gutachter:

Prof. Dr. Günter Rote
Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
Germany
rote@inf.fu-berlin.de

Dr. Remco C. Veltkamp
Department of Information and Computing Sciences
Universiteit Utrecht
Padualaan 14, De Uithof
3584CH Utrecht
The Netherlands
remco.veltkamp@cs.uu.nl

Termin der Disputation: 21. April 2008

Contents

Introduction

Shape matching is an important topic in computational geometry, computer vision, image retrieval, object recognition and robotics. For a fixed distance measure D and a class of transformations $\mathcal{T}$ we can describe the problem as follows: Given two shapes A and B in $\mathbb{R}^d$, find a transformation $T^* \in \mathcal{T}$, such that the distance between A and $T^*(B)$ is minimal among all transformations $T \in \mathcal{T}$. Usually finding such a transformation is computationally expensive, if at all possible. Thus we concentrate on approximation algorithms. We follow an approach by Alt, Behrends and Blömer [4], and Alt, Aichholzer and Rote [3]. They use mappings called reference points to fix the relative position between the two sets. A reference point is a Lipschitz continuous mapping from the set of shapes into $\mathbb{R}^d$ which is equivariant under the considered class of transformations. This approach reduces the degrees of freedom of the underlying problem by the dimension d.

In this thesis we study approximation algorithms for shape matching with respect to various metrics, e.g., the Hausdorff distance, the Earth Mover's Distance, the Monge-Kantorovich Distance and the bottleneck distance. We investigate translations, rigid motions, i.e., combinations of translations and rotations, and similarities, i.e., combinations of rigid motions and scalings.

The basic structure of the approximation algorithms is the same for all metrics and we describe our approach in an abstract reference point framework. We first determine the relative position of the two shapes to each other by computing their reference points. We then translate the shapes such that the reference points coincide. Next we determine a rotation for one of the shapes such that the distance of the two shapes is at most a constant times their optimal distance. Similarities can always be dealt with by finding an approximate scaling before finding the rotation.

In the following we give a short outline of the thesis. The formal definitions are postponed to the introductory sections of the corresponding chapters.

In **Chapter 1** we give a description of shape matching which is suitable for our purposes. We define reference points and prove features of these points. We generalize the approach by Alt, Behrends and Blömer [4], and Alt, Aichholzer and Rote [3], and construct an abstract framework that provides approximation algorithms for translations, rigid motions and similarities. These algorithms can be used for any set of shapes which is closed under the considered class of transformations and for metrics fulfilling the weak property that there is a constant k such that $D(A, \tau(A)) \leq k\|\tau\|$ for any shape A and any translation vector τ.

Chapter 2 is devoted to the Hausdorff distance which is the basis of our reference point framework. We use this chapter to recall the results by Alt, Behrends and Blömer [4], and Alt, Aichholzer and Rote [3], and investigate open questions posed in the latter paper. We prove a lower bound of $1 + \sqrt{1/3} \approx 1.58$ for the approximation ratio of approximation algorithms for translations using reference points.

In **Chapter 3** we discuss weighted point sets and introduce the Earth Mover's Distance (EMD). We show that the center of mass is a reference point for the EMD with respect to affine transformations. For weighted point sets in the plane, we show a 2-approximation algorithm for translations, a 4-approximation algorithm for rigid motions and an 8-approximation algorithm for similarities. The runtime of the translation approximation is $O(T^{\mathrm{EMD}}(n,m))$, the runtime of the latter two algorithms is $O(nm \cdot T^{\mathrm{EMD}}(n,m))$, where $T^{\mathrm{EMD}}(n,m)$ is the time to compute the EMD between two weighted point sets with n and m points, respectively. We also show that these algorithms can be generalized to arbitrary dimension, leading to worse time and approximation bounds.

We indicate that the proven approximation ratios of the algorithms for transformations including the rotations are not tight. In fact, we give a lower bound of approximately 1.155 for the approximation ratio of the algorithm for rotations and argue that this is the true bound. Unfortunately, a formal proof must be postponed to future work.

Parts of this chapter have already been published by Klein and Veltkamp [37, 38].

In **Chapter 4** we introduce the notion of Manhattan networks. A Manhattan network on a set of n points in the plane is a rectlinear network G with the property that for every pair of points in S, the network G contains a path between them whose length equals the Manhattan distance of the two points. A generalization to the d-dimensional space is immediate. In $\mathbb{R}^d$, we show that for any set of n points there is a Manhattan network with $O(n \log^{d-1} n)$ vertices and edges. We reduce the time to compute the Earth Mover's Distance based on the Manhattan metric to $O(n^2 \log^{2d-1} n)$ time. This improves the previously best known result of $O(n^4 \log n)$ significantly. The construction immediately leads to a constant-factor approximation for the Euclidean EMD, which is conceptually easier than the slightly faster $(1+\varepsilon)$-approximation by Cabello, Giannopoulos, Knauer and Rote [15]. The results presented in this chapter are joint work with Gudmundsson, Knauer and Smid [33].

In the following **Chapter 5** we generalize the results gained in Chapter 3 to the Monge-Kantorovich Distance on probability measures. We observe that the algorithm for translations carries over. In contrast, the algorithm to find an approximation for rotations, which is using the discreteness in the case of weighted point sets, cannot be generalized directly. Unfortunately we cannot prove a constant-factor approximation in this case, but we are able to prove an approximation algorithm for bounding the absolute error.

In **Chapter 6** we investigate the application of our reference point framework to point sets of equal size with respect to the bottleneck distance. We see that several reference points for this distance measure exist and investigate approximation algorithms for translations, rotations around a fixed point, rigid motions and similarities. For rotations around a fixed point in the plane we give an exact algorithm with runtime $O(n^{5.5} \log n)$. The main contribution of this chapter is that we show a $(1+\sqrt{2}\,)$-approximation for this problem with runtime $O(n^{2.5} \log n)$. Thus, except for the slightly worse approximation factor, we improve the previously best known result of Agarwal and Phillips [1] by nearly a linear factor. Based on this, we show a $2(1+\sqrt{2}\,)$-approximation for rigid motions and a $4(1+\sqrt{2}\,)$ approximation for similarity transformations with runtime $O(n^{2.5} \log n)$. We show how to use δ-nets to improve the approximation ratio for rotations to $2+\varepsilon$. We further generalize the results to higher dimensions, leading to runtimes and approximation ratios exponential in the dimension. We derive fully polynomial-time approximation schemes by standard discretization methods for translations and rigid motions in the plane. The dependence on ε^{-1} is quadratic in the first case and cubic in the second case.

In the final **Chapter 7** we give a short survey on similar shape matching approaches which can be found in the literature. These concern the Fréchet Distance, the volume of symmetric difference, the volume of overlap and the Frobenius norm.

Acknowledgements Many people directly or indirectly contributed to this thesis. I want to thank all of them equally, independently of the order in the following statement.

The first person I want to mention is my advisor Günter Rote. His remarks and comments during inspiring discussions brought light into the darkest holes and his final reading helped a lot to improve the writing and the results of the thesis.

I also want to thank Remco Veltkamp for being my advisor during my research stay in Utrecht. The intensive work with him introduced me to another way of thinking and an efficient way of working.

Special thanks go to Maike Buchin and Astrid Sturm for reading almost every part of this thesis. Their critical reading and accurate corrections polished the work to the final style.

For fruitful discussions on various parts of this thesis I want to thank Helmut Alt, Christian Knauer and Tobias Lenz.

I want to thank the professors and fellows of the European Graduate Program "Combinatorics, Geometry and Computation". The scientific environment in Berlin and the stays abroad provided an optimal basis for my research and the opportunity to write this thesis.

I thank all former and current colleagues in the work group "Theoretische Informatik" at the Freie Universität Berlin. Going to office in the morning always felt like meeting a couple of friends and sharing a nice day.

Special thanks to Claudia Dieckmann for sharing the office for a very long time, Frank Hoffmann for personal advice, Klaus Kriegel for taking care of my motorbike in the winter, the "Sturms" for taking care of me, and Andrea Hoffkamp and Tamara Knoll for their patient help in all organizational duties.

Thanks to Mark Overmars, Marc van Kreveld, Remco Veltkamp and their work group for being a host during my research stay in the Netherlands. I still think about my nice days in Utrecht where I not only made some nice research but also made some nice friends.

I want to thank all my coauthors Joachim Gudmundsson, Christian Knauer, Tobias Lenz, Michiel Smid and Remco Veltkamp. Not only the result, but also their knowledge and experience they shared with me found their way into this thesis.

Finally, I want to thank my parents, my family, my friends and Dagmar. Without their constant support the long tramp to the creation of this thesis would have been abandoned years ago.

Berlin, July 10, 2008

Chapter 1

The General Reference Point Framework

In this chapter we give a short introduction to shape matching and constant-factor approximation algorithms. We further state what we mean by computing the optimal distance under a given class of transformations. We introduce reference points and construct abstract approximation algorithms to find the optimum. These algorithms are independent of the concrete choice of the set of shapes or the metric on this set.

1.1 Shape Matching

Shape matching is an important topic in computational geometry, computer vision, image retrieval, object recognition and robotics. One of the main tasks is to decide: Given two shapes, how much do they resemble each other?

In this thesis we work on different classes of shapes in $\mathbb{R}^d$, for example, arbitrary compact subsets, weighted point sets, and point sets with a fixed number of points. Furthermore we generalize our approach to Borel probability measures, which can be interpreted as an abstract type of shapes, see Section 5.1. As stated before, this chapter is independent of a concrete choice of the set of shapes. We denote an arbitrary set of shapes in $\mathbb{R}^d$ by $\mathcal{S}$.

Finding a good measure of resemblance is a difficult task and cannot be done without knowledge of the application. A good choice of a resemblance function for one application may be a bad one for another. In this thesis we talk about different well-known metrics on shapes, i.e., the Hausdorff distance, the Earth Mover's Distance, the Monge-Kantorovich Distance and the bottleneck distance.

In the following we develop a shape matching framework for arbitrary sets of shapes. This framework can be applied whenever the resemblance function D is a metric. That is, we assume that the mapping $D\colon \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ fulfills the following conditions:

- $\forall A, B \in \mathcal{S} \,:\, D(A,B) \geq 0$ and $D(A,B) = 0 \Leftrightarrow A = B$ (positive definite)
- $\forall A, B \in \mathcal{S} \,:\, D(A,B) = D(B,A)$ (symmetry)
- $\forall A, B, C \in \mathcal{S} \,:\, D(A,B) \leq D(A,C) + D(C,A)$ (triangle inequality).

In many applications the closest resemblance has to be determined where one of the objects is allowed to be transformed in a certain way. Typical classes of transformations under consideration are translations, rigid motions and similarities. Rigid motions are combinations of translations and rotations, and similarities are combinations of rigid motions and scalings. In this work we only consider orientation preserving transformations. The treatment of reflections is easy by determining the resemblance twice, once between the original sets and once, where one of the two sets is reflected.

Let $\mathcal{T}$ be a considered set of transformations in $\mathbb{R}^d$. We will always assume that the set of shapes $\mathcal{S}$ is closed under $\mathcal{T}$. Let $A, B \in \mathcal{S}$ be two shapes. The minimum distance D under transformations with respect to $\mathcal{T}$ is defined as

$$D^{\text{opt}}(A, B) \;:=\; \min_{T \in \mathcal{T}} \{D(A, T(B))\}\,.$$

We also say that we minimize D under $\mathcal{T}$.

Typically, the calculation of the exact solution is computationally expensive, if it is at all possible. Therefore we concentrate on approximation algorithms. By an approximation algorithm we usually mean a constant-factor approximation, i.e., for two shapes $A, B \in \mathcal{S}$ we want to find a transformation $T \in \mathcal{T}$, such that

$$D(A, T(B)) \;\leq\; \alpha \cdot D^{\text{opt}}(A, B),$$

where $\alpha \geq 1$ is a fixed constant independent of A and B. In this case we call α the approximation factor or approximation ratio of the algorithm. The advantages of constant-factor approximations over algorithms which bound the absolute error are immediate. In the case of exact matches, i.e., A equals B, constant-factor approximations are guaranteed to find the exact result. Even in the case when the two shapes do not match exactly but are very similar, a constant-factor approximation is forced to find a good match.

1.2 Weak Reference Points

A weak reference point is an equivariant mapping from the set of shapes into $\mathbb{R}^d$. It has the property that for any two shapes there is a transformation T in the considered set of transformations $\mathcal{T}$ for which the two weak reference points coincide and the distance of the two transformed figures is at most a constant factor times the optimal distance. This structure was first defined by Alt, Behrends and Blömer [4].

Definition 1.1 (Weak Reference Point)**.** [4] Let $\mathcal{S}$ be a set of shapes in $\mathbb{R}^d$ and D a metric on $\mathcal{S}$. A mapping $r\colon \mathcal{S} \to \mathbb{R}^d$ is called a weak D-reference point for $\mathcal{S}$ with respect to $\mathcal{T}$ if the following two conditions hold:

1. Equivariance with respect to $\mathcal{T}$: For all $A \in \mathcal{S}$ and $T \in \mathcal{T}$ we have
$$r(T(A)) \;=\; T(r(A)).$$
2. Approximation property: For all $A, B \in \mathcal{S}$ there is a transformation $T \in \mathcal{T}$ with $r(A) = r(T(B))$, such that
$$D(A, T(B)) \;\leq\; \alpha \cdot D^{\text{opt}}(A, B).$$

The constant α is called the approximation factor of the weak reference point r.

Alt, Behrends and Blömer [4] call this structure a reference point. In our work, a reference point is a mapping which is equivariant under a class of transformations, and which is Lipschitz continuous, see Definition 1.2. As we will see later, this definition is more restrictive. Therefore we call the structure defined by Alt, Behrends and Blömer a weak reference point.

1.3 Reference Points

A reference point is a Lipschitz continuous mapping which is equivariant under the considered class of transformations. These points have been introduced by Alt, Aichholzer and Rote [3] to construct approximation algorithms for matching compact subsets of $\mathbb{R}^d$ under translations, rigid motions and similarities with respect to the Hausdorff distance. More results concerning reference points for the Hausdorff and Fréchet distance can be found in the Ph.D. theses by Knauer [39] and Wenk [52]. We give a generalized definition suitable for the construction of our shape matching framework for different classes of shapes and metrics.

Definition 1.2 (Reference Point)**.** [3, 39] Let $\mathcal{S}$ be a set of shapes and $D\colon \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ be a metric on $\mathcal{S}$. A mapping $r\colon \mathcal{S} \to \mathbb{R}^d$ is called a D-reference point for $\mathcal{S}$ with respect to a set of transformations $\mathcal{T}$, if the following two conditions hold:

1. Equivariance with respect to $\mathcal{T}$: For all $A \in \mathcal{S}$ and $T \in \mathcal{T}$ we have

$$r(T(A)) \;=\; T(r(A)).$$

2. Lipschitz continuity: There is a constant $c \geq 0$, such that for all $A, B \in \mathcal{S}$ we have

$$\|r(A) - r(B)\| \;\leq\; c \cdot D(A, B).$$

We call the smallest constant c such that condition 2 holds, the Lipschitz constant of the D-reference point r. The unspecified norm $\|\cdot\|\colon \mathbb{R}^d \to \mathbb{R}$ is called the underlying norm on $\mathbb{R}^d$.

In Section 1.5.1 we see that under an additional condition on the metric D, a reference point is always a weak reference point. However, it is still open if there are weak reference points which are not Lipschitz continuous.

1.4 Properties of Reference Points

In the following chapters we consider various metrics on different classes of shapes in $\mathbb{R}^d$. For every metric and class of shapes we find reference points and prove their Lipschitz constants. In the next few sections we show how we can use reference points to construct more reference points for the same metric, or show that the same point is also a reference point for another metric.

1.4.1 Translated Reference Points

Given a reference point with respect to translations, one can add every constant vector to this reference point to obtain a new one. Alt, Aichholzer and Rote [3] use this in a special case. We generalize the result:

Theorem 1.1. *[3] Let $\mathcal{S}$ be a set of shapes and D a metric on $\mathcal{S}$. Let $r\colon \mathcal{S} \to \mathbb{R}^d$ be a D-reference point with respect to translations. Then $r'\colon \mathcal{S} \to \mathbb{R}^d, A \mapsto r(A) + v$, where $v \in \mathbb{R}^d$ is any fixed vector, is a D-reference point with respect to translations. The Lipschitz constant c' of r' is equal to the Lipschitz constant c of r.*

Proof. We have to show the equivariance and the Lipschitz continuity of r':

1. Equivariance:
   Let τ be any translation and let τ_v be the translation by v. Then

$$r'(\tau(A)) \;=\; \tau_v(r(\tau(A))) \;=\; \tau_v(\tau(r(A))) \;=\; \tau(\tau_v(r(A))) \;=\; \tau(r(\tau_v(A))) \;=\; \tau(r'(A)).$$

2. Lipschitz continuity:

$$\|r'(A) - r'(B)\| \;=\; \|r(A) + v - (r(B) + v)\| \;=\; \|r(A) - r(B)\| \;\leq\; c \cdot D(A, B).$$

The lower bound for the Lipschitz constant of r' is given by the same pair of shapes $A, B \in \mathcal{S}$ which prove the lower bound for r. ☐

Thus there are infinitely many D-reference points with respect to translations. However, these reference points are not really distinct because in our approximation algorithm they lead to the same relative position of the sets with respect to each other and therefore lead to the same value of the approximate distance D, see Section 1.5.

So, the above result might seem useless at first. However, suppose the set of shapes $\mathcal{S}$ includes single points $x \in \mathbb{R}^d$. Then, by Theorem 1.1 we can always assume that $r(\{x\}) = x$. If this does not hold, let O denote the origin of $\mathbb{R}^d$. We subtract the constant vector $r(\{O\})$ from r, meaning $r'(A) := r(A) - r(\{O\})$. Thus, $r'(\{O\}) = O$. Then, $r'(\{x\}) = r'(O + x) = r'(\{O\}) + x = x$. Therefore we can always assume that $r(\{x\}) = x$ for all $x \in \mathbb{R}^d$, which seems to be a natural assumption. Note that this property is forced if the reference point is equivariant under rotations.

1.4.2 Convex Combination of Reference Points

It is advantageous to use several reference points and thereby obtain several approximating positions of the sets. In this section we prove that we can construct a whole family of reference points by choosing any convex combination of a set of given reference points. An upper bound on the Lipschitz constant of this reference point is given by the same convex combination of the Lipschitz constants of the used reference points. Though we do not have any example of a convex combination of reference points where its Lipschitz constant does not match the upper bound, the convex combination of a large number of reference points might lead to a better approximation in practical applications. The result holds for any class of transformations contained in the class of affine mappings. This includes translations, rigid motions and similarities. Alt, Aichholzer and Rote [3] use this in a special case. We generalize the result:

Theorem 1.2. *[3] Let $\mathcal{S}$ be a set of shapes in $\mathbb{R}^d$ and D a metric on $\mathcal{S}$. Let $\mathcal{T}$ be a subset of the class of affine mappings. Let $r_1, \ldots, r_m\colon \mathcal{S} \to \mathbb{R}^d$ be D-reference points with respect to $\mathcal{T}$ and let $c_1, \ldots, c_m$ denote their Lipschitz constants. Every convex combination*

$$r \;:=\; \sum_{j=1}^{m} \lambda_j r_j, \qquad \text{with} \qquad 0 \;\leq\; \lambda_j \;\leq\; 1 \qquad \text{and} \qquad \sum_{j=1}^{m} \lambda_j \;=\; 1$$

is a D-reference point with respect to $\mathcal{T}$. Its Lipschitz constant is at most $\sum_{i=1}^{m} \lambda_j c_j$.

Proof. We have to show the equivariance of the convex combination under affine mappings and its Lipschitz continuity. We describe an arbitrary affine mapping by $M+\tau$, where M describes a linear transformation in $\mathbb{R}^d$ and τ, with a slight abuse of notation, denotes the translation as well as the translation vector.

- Equivariance:

$$\begin{aligned}
r((M+\tau)(A)) &= \sum_{i=1}^{m} \lambda_i r_i((M+\tau)(A)) \\
&= \sum_{i=1}^{m} \lambda_i (M+\tau)(r_i(A)), \qquad \text{by the equivariance of } r_i \\
&= \sum_{i=1}^{m} \lambda_i M(r_i(A)) + \sum_{i=1}^{m} \lambda_i \tau(r_i(A)) \\
&= M(r(A)) + \sum_{i=1}^{m} \lambda_i (r_i(A)) + \tau), \qquad \text{by linearity of } M \\
&= M(r(A)) + \sum_{i=1}^{m} \lambda_i r_i(A) + \sum_{i=1}^{m} \lambda_i \tau \\
&= M(r(A)) + r(A) + \tau, \qquad \text{since } \textstyle\sum_{i=1}^{m} \lambda_i = 1 \\
&= M(r(A)) + \tau(r(A)) \\
&= (M+\tau)(r(A)).
\end{aligned}$$

- Lipschitz continuity:

$$\begin{aligned}
\|r(A) - r(B)\| &= \left\| \sum_{i=1}^{m} \lambda_i r_i(A) - \sum_{i=1}^{m} \lambda_i r_i(B) \right\| \\
&\leq \sum_{i=1}^{m} \lambda_i \cdot \|r_i(A) - r_i(B)\| \\
&\leq \sum_{i=1}^{m} \lambda_i c_i \cdot D(A,B).
\end{aligned}$$

□

1.4.3 Extension of Reference Points to Other Metrics

We show that the existence of a D-reference point can be extended to the existence of a D'-reference point, if the metric D can be bounded by a multiple of D' from above. The proof is easy but it often enables us to find new reference points. In a special case this was already used by Knauer [39].

Theorem 1.3. *[39] Let $\mathcal{S}$ be a set of shapes in $\mathbb{R}^d$, $\mathcal{T}$ a set of transformations, and D a metric on $\mathcal{S}$. Let $r\colon \mathcal{S} \to \mathbb{R}^d$ be a D-reference point with respect to $\mathcal{T}$ and with Lipschitz constant c. Let $D'\colon \mathcal{S} \to \mathbb{R}^d$ be another metric on $\mathcal{S}$ with $D(A,B) \leq k \cdot D'(A,B)$ for any two shapes*

$A, B \in \mathcal{S}$ and a fixed constant $k \in \mathbb{R}_{>0}$. Then r is a D'-reference point with respect to $\mathcal{T}$ and Lipschitz constant at most ck.

Proof. Since the equivariance of r is independent of the choice of the metric, we only need to prove the Lipschitz continuity. This follows easily by

$$\|r(A) - r(B)\| \;\leq\; c \cdot D(A, B) \;\leq\; ck \cdot D'(A, B)$$

for any two shapes $A, B \in \mathcal{S}$. □

1.4.4 Lower Bound for the Lipschitz Constant of a Reference Point

Later we construct abstract approximation algorithms using reference points. The approximation ratios of these algorithms depend on the Lipschitz constant of the used reference point. We prove a lower bound for the Lipschitz constant of a reference point if the metric on the set of shapes fulfills a weak condition. Then, this lower bound for the Lipschitz constant immediately gives us a lower bound for the approximation ratio which can be proven using our approach. Therefore it can be used to estimate the quality of such an algorithm.

Theorem 1.4. *Let $\mathcal{S}$ be a set of shapes in $\mathbb{R}^d$ and $\mathcal{T}$ a set of transformations including the translations. Let the metric D on $\mathcal{S}$ fulfill the condition*

$$D(A, \tau(A)) \;\leq\; \|\tau\|$$

for any shape $A \in \mathcal{S}$ and translation vector $\tau \in \mathbb{R}^d$. Let $r\colon \mathcal{S} \to \mathbb{R}^d$ be a D-reference point with respect to $\mathcal{T}$ and let c be its Lipschitz constant. Then $c \geq 1$.

Proof. Let $A \in \mathcal{S}$ be an arbitrary shape and let $B := A + \tau$ where $\tau \neq 0$ is a translation vector. By the Lipschitz continuity of r we have

$$\|r(A) - r(B)\| \;\leq\; c \cdot D(A, B) \;\leq\; c\,\|\tau\|.$$

On the other hand, using the equivariance of r with respect to translations we get

$$\|r(A) - r(B)\| \;=\; \|r(A) - (r(A) + \tau)\| \;=\; \|\tau\|.$$

Since $\|\tau\| \neq 0$ the claim follows. □

1.5 Abstract Approximation Algorithms

Later we use reference points to construct approximation algorithms for shape matching. We derive methods for various metrics and different sets of shapes. In this section we show abstract algorithms independent of a concrete choice of the metric or class of shapes. Some of them have already been mentioned by Alt, Aichholzer and Rote [3]. These abstract algorithms form a general framework for shape matching using reference points. The framework for two shapes $A, B \in \mathcal{S}$ works as follows: We first translate B such that the reference points of the two sets coincide. We then compute an optimal or approximate rotation and scaling of the translated version of B around their coinciding reference points. Thereby we obtain approximation algorithms for translations, rigid motions and similarities. The first step, namely putting the reference points on top of each other and fixing this point as the rotation center, reduces the degrees of freedom of the underlying matching problems and therefore yields efficient algorithms.

For the remainder of the chapter let $A, B \in \mathcal{S}$ be two shapes. We use $T^{\mathrm{ref}}(A)$ to denote the time to compute the reference point of A. We further use $T^D(A, B)$ to denote the time to compute the metric D between A and B.

1.5.1 Approximation for Translations

The abstract algorithm for matching under translations can be described as follows:

Algorithm 1.1.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.
2. Output B' together with the distance $D(A, B')$.

We prove that Algorithm 1.1 leads to a constant-factor approximation if the metric D on the shapes is well-behaved in some sense. Note that this condition is completely independent of the considered reference point.

Theorem 1.5. *Let $r\colon \mathcal{S} \to \mathbb{R}^d$ be a D-reference point with respect to translations. Let c be its Lipschitz constant. Let $k \in \mathbb{R}_{>0}$ be some constant and let D fulfill the condition*

$$D(A, \tau(A)) \;\leq\; k \, \|\tau\|$$

for any shape $A \in \mathcal{S}$ and translation vector $\tau \in \mathbb{R}^d$. Then Algorithm 1.1 induces a constant-factor approximation for D under translations with approximation factor $1 + kc$. Its runtime is $O(T^{\mathrm{ref}}(A) + T^{\mathrm{ref}}(B) + T^D(A, B))$.

Proof. Let τ^{opt} denote an optimal translation of B. Let $\tau^{\mathrm{ref}} := r(A) - r(B)$ be the translation for which the two reference points coincide. Note that this is the translation given by Algorithm 1.1. Further, $\tau := r(A) - r(\tau^{\mathrm{opt}}(B))$ denotes the translation for which the two reference points of A, and B in optimal position coincide. Then,

$$\begin{aligned}
D(A, \tau^{\mathrm{ref}}(B)) &= D(A, \tau(\tau^{\mathrm{opt}}(B))) \\
&\leq D(A, \tau^{\mathrm{opt}}(B)) + D(\tau^{\mathrm{opt}}(B), \tau(\tau^{\mathrm{opt}}(B))) \\
&\leq D(A, \tau^{\mathrm{opt}}(B)) + k \cdot \|\tau\| \\
&= D(A, \tau^{\mathrm{opt}}(B)) + k \cdot \|r(A) - r(\tau^{\mathrm{opt}}(B))\| \\
&\leq D(A, \tau^{\mathrm{opt}}(B)) + kc \cdot D(A, \tau^{\mathrm{opt}}(B)),
\end{aligned}$$

where the last inequality follows by the Lipschitz continuity of the reference point.

The runtime of the algorithm depends on the time to compute the D-reference points, and the time to compute the distance between A and B. □

1.5.2 Fully Polynomial-Time Approximation Scheme for Translations

Efrat, Itai and Katz [25] use Algorithm 1.1 to construct a $(1+\varepsilon)$-approximation algorithm for the bottleneck distance under translations. We generalize their construction to obtain an abstract $(1+\varepsilon)$-approximation algorithm.

A closer look at the proof by Efrat et al. [25] shows that the approach depends on the knowledge of some value $\alpha \in \mathbb{R}_+$ with $D_{\mathcal{B}}(A, \tau^{\text{opt}}(B)) \leq \alpha \leq (1+\beta) \cdot D_{\mathcal{B}}(A, \tau^{\text{opt}}(B))$ for some fixed constant $\beta \in \mathbb{R}_{>0}$, where $D_{\mathcal{B}}$ denotes the bottleneck distance, see Chapter 6, and τ^{opt} denotes the optimal translation. Hence we can use any D-reference point and apply kc as β, see Theorem 1.5. To prove this result we follow the notation and technique of Efrat et al. [25]

We prove the theorem for all L_p-distances with $1 \leq p \leq \infty$. Recall that all norms in $\mathbb{R}^d$ are equivalent. Therefore the result can be extended to any norm on $\mathbb{R}^d$ by adjusting the grid size appropriately.

Let $\text{diam}(p, d)$ denote the diameter of the d-dimensional unit cube in the underlying L_p-norm, i.e., for finite p, $\text{diam}(p, d) = \sqrt[p]{d}$ and $\text{diam}(\infty, d) = 1$.

Theorem 1.6. *[25] Let $1 \leq p \leq \infty$. Let $\mathcal{S}$ be a class of shapes and D a metric on $\mathcal{S}$. Let $A,\ B \in \mathcal{S}$ and let $r\colon \mathcal{S} \to \mathbb{R}^d$ be a D-reference point with respect to translations and with Lipschitz constant c. Let D fulfill the condition*

$$D(A, \tau(A)) \ \leq \ k \, \|\tau\|_p$$

for any shape $A \in \mathcal{S}$, translation vector $\tau \in \mathbb{R}^d$ and some constant $k \in \mathbb{R}_{>0}$. Then there exists an algorithm that for any $0 < \varepsilon < 1$ finds a translation τ^{ε}, such that

$$D(A, \tau^{\varepsilon}(B)) \ \leq \ (1+\varepsilon) \cdot D(A, \tau^{\text{opt}}(B)).$$

Its runtime is $O(\varepsilon^{-d} \cdot T^D(A, B) + T^{\text{ref}}(A) + T^{\text{ref}}(B))$.

Proof. Let $0 < \varepsilon < 1$ and $r\colon \mathcal{S} \to \mathbb{R}^d$ be any D-reference point with respect to translations. Let $\tau^{\text{ref}} := r(A) - r(B)$ be the translation of B for which the two reference points of A and B coincide. Let $\alpha := D(A, \tau^{\text{ref}}(B))$ be the approximate distance computed by Algorithm 1.1, i.e., $D(A, \tau^{\text{opt}}(B)) \leq \alpha \leq (1+kc) \cdot D(A, \tau^{\text{opt}}(B))$. Let cube$[c\alpha]$ denote the axis-parallel cube of side length $2c\alpha$ centered at τ^{ref}. Consider a grid Γ centered at τ^{ref} with cell size $\gamma := \varepsilon\alpha/(\text{diam}(p, d)k(1+kc))$. The distance of any point of cube$[c\alpha]$ to its closest grid point of Γ is at most $\gamma \cdot \text{diam}(p, d) = \varepsilon\alpha/k(1+kc)$. See Figure 1.1 for the construction.

We first have to show that τ^{opt} is inside the cube:

$$\begin{aligned}
\left\|\tau^{\text{opt}} - \tau^{\text{ref}}\right\|_p &= \left\|\tau^{\text{opt}} - (r(A) - r(B))\right\|_p \\
&= \left\|r(A) - (r(B) + \tau^{\text{opt}})\right\|_p \\
&\leq c \cdot D(A, \tau^{\text{opt}}(B)) \\
&\leq c\alpha.
\end{aligned}$$

Now, let τ^{ε} be a translation minimizing $D(A, B+\tau)$ over all grid points $\tau \in \mathbb{R}^d$ inside the cube.

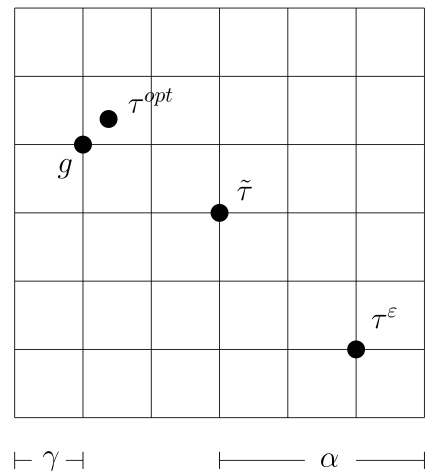


Figure 1.1: Construction for the $(1+\varepsilon)$-approximation algorithm.

Let g be a grid point closest to τ^{opt}. We show that $D(A, \tau^{\varepsilon}(B)) \leq (1+\varepsilon) \cdot D(A, \tau^{\text{opt}}(B))$.

$$\begin{aligned}
D(A, \tau^{\varepsilon}(B)) &\leq D(A, g(B)) \\
&\leq D(A, \tau^{\text{opt}}(B)) + D(\tau^{\text{opt}}(B), g(B)) \\
&= D(A, \tau^{\text{opt}}(B)) + D(B, (g - \tau^{\text{opt}})(B)) \\
&\leq D(A, \tau^{\text{opt}}(B)) + k \cdot \|g - \tau^{\text{opt}}\|_p \\
&\leq D(A, \tau^{\text{opt}}(B)) + k\gamma \cdot \text{diam}(p, d) \\
&\leq D(A, \tau^{\text{opt}}(B)) + \frac{\varepsilon\alpha}{1+kc} \\
&\leq D(A, \tau^{\text{opt}}(B)) + \frac{\varepsilon}{1+kc}(1+kc) \cdot D(A, \tau^{\text{opt}}(B)).
\end{aligned}$$

The number of grid points inside the cube is

$$(1 + \frac{2c\alpha}{\gamma})^d = O((1+\varepsilon^{-1})^d).$$

The runtime is given by the time to compute the reference points of A and B, and the time to compute their distance D at $O(\varepsilon^{-d})$ grid points. □

1.5.3 Approximation for Rigid Motions

In the following we investigate the problem to compute the minimum distance D of two shapes under rigid motions. Since this does not make sense in one dimension, we always assume that the dimension d is greater than or equal to 2. The following algorithm computes an approximation of D under rigid motions. For a point $p^* \in \mathbb{R}^d$ let $\text{Rot}(p^*)$ denote the set of rotations around p^*. The time and method to find the rotation in Step 2 depends on the set of shapes and the metric, and will be addressed in the corresponding chapters.

Algorithm 1.2.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.

2. Determine a rotation $R' \in \mathrm{Rot}(r(A))$, such that

$$D(A, R'(B')) \;\leq\; \rho \cdot \min_{R \in \mathrm{Rot}(r(A))} D(A, R(B)).$$

3. Output $R'(B')$ together with the distance $D(A, R'(B'))$.

Let T^{rot} denote the time to compute the rotation in step 2.

Theorem 1.7. *Let $\mathcal{S}$ be a class of shapes and D a metric on $\mathcal{S}$. Let $A, B \in \mathcal{S}$ and let $r\colon \mathcal{S} \to \mathbb{R}^d$ be a D-reference point with respect to rigid motions and with Lipschitz constant c. Let D fulfill the condition*

$$D(A, \tau(A)) \;\leq\; k\,\|\tau\|$$

for any shape $A \in \mathcal{S}$, translation vector $\tau \in \mathbb{R}^d$ and some constant $k \in \mathbb{R}_{>0}$. Then, Algorithm 1.2 finds an approximately optimal matching for rigid motions with approximation factor $\rho(1+kc)$ in time $O(T^{\mathrm{ref}}(A) + T^{\mathrm{ref}}(B) + T^{D}(A,B) + T^{\mathrm{rot}})$.

Proof. Let $A, B \in \mathcal{S}$ be two arbitrary shapes. Let M^{opt} be the rigid motion minimizing $D(A, M(B))$ over all rigid motions M. Let $\tau := r(A) - r(M^{\mathrm{opt}}(B))$ be the translation for which the reference points of A and B in optimal position coincide. Let M^* be a rigid motion minimizing $D(A, M(B))$ while mapping $r(B)$ onto $r(A)$. Let $\tau^{\mathrm{ref}} := r(A) - r(B)$ be the translation moving B in a way that its reference point coincides with the reference point of A. Let R' be the approximate rotation determined in Algorithm 1.2. Then we have

$$\begin{aligned}
D(A, R' \circ \tau^{\mathrm{ref}}(B)) &\leq \rho \cdot \min_{R \in \mathrm{Rot}(r(A))} D(A, R \circ \tau^{\mathrm{ref}}(B)) \\
&= \rho \cdot D(A, M^*(B)) \\
&\leq \rho \cdot D(A, \tau \circ M^{\mathrm{opt}}(B)) \\
&\leq \rho \cdot (D(A, M^{\mathrm{opt}}(B)) + D(M^{\mathrm{opt}}(B), \tau \circ M^{\mathrm{opt}}(B))) \\
&\leq \rho \cdot (D(A, M^{\mathrm{opt}}(B)) + k \cdot \|\tau\|) \\
&\leq \rho \cdot (D(A, M^{\mathrm{opt}}(B)) + kc \cdot D(A, M^{\mathrm{opt}}(B))) \\
&= \rho(1+kc) \cdot D(A, M^{\mathrm{opt}}(B)).
\end{aligned}$$

The runtime of this algorithm depends on the time to compute the D-reference points, translate B such that the D-reference points coincide, find the rotation R' of the translated version of B around $r(A)$, and compute the distance D between A and the translated version of B rotated by R'. □

Fixing the position of the coinciding D-reference points as the rotation center in Algorithm 1.2 eliminates several degrees of freedom and the problem to find the optimal rotation is easier than the one finding the optimal rigid motion itself. For various distance measures, however, even for this problem there is no efficient algorithm known so far. Therefore we investigate approximation algorithms. These algorithms are often based on computing the distance D for each rotation where two event points given by the shapes are aligned. By this we mean that those event points ly on the same ray starting at the rotation center.

1.5.4 Approximation for Similarities

The problem to find a similarity minimizing the distance between two shapes A and B differs from the problems to find the optimal translation or rigid motion. In the latter two cases it makes no difference, if A, or B, or both A and B are allowed to be transformed. The optimal position of the two shapes depends on the choice which shapes are transformed, but in contrast, the optimal distance is the same for every choice.

In the case of similarities the situation is different. If we allow both sets to be transformed, the optimal distance will be zero by scaling both sets with a scaling factor of zero. If we allow only one set to be transformed, the optimal distance will depend on the choice which one to reshape. Another approach which can be found in the literature is bounding the scaling factor from below.

In our interpretation we leave one of the shapes fixed and the other is free to be transformed by any similarity. Which one to be reshaped will be clear from the context. Thus we want to compute $\min_S D(A, S(B))$, where the minimum is taken over all similarities S. We only consider positive similarities, i.e., the scaling factor is positive. The consideration of negative similarities is easy by using the same algorithms on B and a reflected copy of A.

Basically, the approach is to use the algorithm for rigid motions on the two shapes, where B is scaled by some value α. This value depends on the metric. For example, in the case of the EMD we scale by the quotient of the normalized first moments, see Section 3.4.5; in the case of the bottleneck distance we scale by the quotient of the distances of the furthest points to the reference point, see Section 6.11.

We state the abstract algorithm for similarities below. We prove the approximation ratios and runtimes in the appropriate chapters.

Algorithm 1.3.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.

2. Determine an approximate scaling factor α,
   and scale B' by α around the center $r(A)$.
   Let B'' be the image of B' under this scaling.

3. Determine a rotation $R' \in \mathrm{Rot}(r(A))$, such that

$$D(A, R'(B')) \;\leq\; \rho \cdot \min_{R \in \mathrm{Rot}(r(A))} D(A, R(B)).$$

4. Output B''' together with the distance $D(A, B''')$.

Chapter 2

The Hausdorff Distance

An important and widely studied distance measure for determining the resemblance of sets in Euclidean space, especially in $\mathbb{R}^2$ and $\mathbb{R}^3$, is the so-called Hausdorff distance. This distance measure can be defined in arbitrary dimension d for the set $\mathcal{C}^d$ of all non-empty compact subsets of $\mathbb{R}^d$ as follows:

Definition 2.1 (Hausdorff Distance)**.** For $A, B \in \mathcal{C}^d$ let

$$\vec{D}_{\mathcal{H}}(A,B) \;:=\; \max_{a\in A}\min_{b\in B}\|a-b\|_2$$

be the directed Hausdorff distance from A to B. The Hausdorff distance between A and B is defined as the maximum of the directed Hausdorff distances from A to B and from B to A:

$$D_{\mathcal{H}}(A,B) \;:=\; \max\{\vec{D}_{\mathcal{H}}(A,B), \vec{D}_{\mathcal{H}}(B,A)\}.$$

Example. In Figure 2.1 we give an example for the directed Hausdorff distances between two convex figures. The Hausdorff distance is equal to $\vec{D}_{\mathcal{H}}(B,A)$.

The Hausdorff distance defines a metric on $\mathcal{C}^d$ and is defined on a very general set of shapes. Therefore it has a wide range of applications and several exact and approximation algorithms to compute the distance have been given. Most of these algorithms concentrate on shapes in $\mathbb{R}^2$ and $\mathbb{R}^3$. For two fixed disjoint convex sets in the plane, Atallah [11] gives a linear time algorithm based on rotating calipers. An $O((n+m)\log(n+m))$ time method is known for

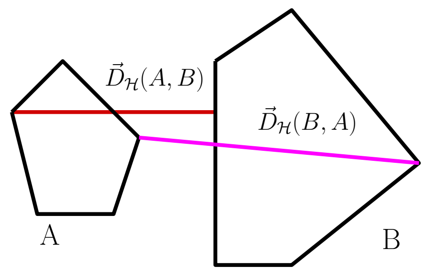


Figure 2.1: The Hausdorff distance between two convex objects in $\mathbb{R}^2$.

the case that the sets A and B consist of n and m line segments in the plane, respectively, see the work of Alt, Behrends and Blömer [4]. This algorithm is based on the computation of the Voronoi diagrams of the two sets.

However, it is more natural to assume that A and B are not static, but can be transformed by a given set of transformations. In this case it is interesting to find the optimal transformation which minimizes the Hausdorff distance. Several algorithms are known to solve this problem exactly. For the two dimensional problem Alt, Behrends, and Blömer [4] give an $O((nm)\log(nm)\log^*(nm))$ time algorithm for the set of translations along a fixed direction; Agarwal, Sharir, and Toledo [2] describe an algorithm for arbitrary translations with runtime $O((nm)^2\log^3(nm))$. This method can be improved to $O((nm)^2\alpha(nm))$ time if A and B are finite sets of points, see Huttenlocher and Kedem [35]. Chew, Goodrich, Huttenlocher, Kedem, Kleinberg and Kravets [17] give an algorithm for minimizing the Hausdorff distance under rigid motions. The runtime of their method is $O((nm)^3\log^2(nm))$. The latter two algorithms use sophisticated and powerful tools like parametric search.

Alt, Aichholzer and Rote [3] follow the approach to approximate the result using reference points. They define reference points for the Hausdorff distance and use these points to derive approximation algorithms for translations, rigid motions and similarities. In other words they define the basis for our general reference point framework introduced in Chapter 1. Let r be a reference point with respect to the considered class of transformations and with Lipschitz constant c. Then, the approximation ratios of their algorithms are $1+c$ for translations and rigid motions, and $3+c$ for similarities. In the plane, the runtimes for two sets of n and m points and line segments are $O((n+m)\log(n+m))$ for translations and $O(nm\log(nm)\log^*(nm))$ for rigid motions and similarities. In $\mathbb{R}^3$, where the two sets consist of n and m triangles, the runtimes become $O(nm)$ for translations and $O((nm)^3 \cdot T^{D_\mathcal{H}}(n,m))$ for rigid motions and similarities, where $T^{D_\mathcal{H}}(n,m)$ denotes the time to compute the Hausdorff distance in $\mathbb{R}^3$.

2.1 Results

In this chapter we derive lower bounds concerning shape matching with respect to the Hausdorff distance in the plane.

First, we give a lower bound for the Lipschitz constant of the Steiner point, see Definition 2.2. Alt, Aichholzer and Rote [3] already show two compact subsets in the plane proving this bound, where one of the sets is non-convex. Our lower bound consists of convex sets only.

Second, we give a concrete set of convex sets in the plane which proves that the Lipschitz constant of any reference point has to be greater than or equal to $\sqrt{4/3}$. Such an example is also given by Alt, Aichholzer and Rote [3]. But again, in contrast to our lower bound, their example consists of non-convex sets.

Finally, we give convex sets in the plane which prove a lower bound for the abstract algorithm for matching under translations, see Section 1.5.1. This lower bound is independent of a concrete choice of a $D_\mathcal{H}$-reference point. In a first approach, these sets have been found using a computer program. We prove that the approximation ratio is at least $1+\sqrt{1/3}$.

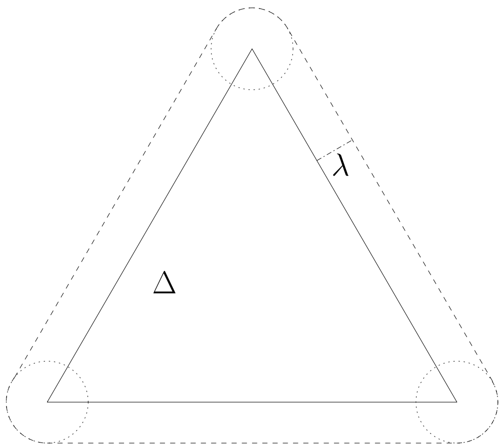


Figure 2.2: The λ-neighbourhood $U_\lambda(\Delta)$.

2.2 Preliminaries

The following lemma gives a commonly-known reformulation of the Hausdorff distance, which is important for our computer experiments, see Section 2.6.

Lemma 2.1. *Let $A, B \in \mathcal{C}^d$. Then*

$$D_{\mathcal{H}}(A,B) \;=\; \min\left\{\lambda \geq 0 \mid A \subset U_\lambda(B) \text{ and } B \subset U_\lambda(A)\right\},$$

where $U_\lambda(A) = \{x \in \mathbb{R}^d \mid \exists a \in A \text{ s.t. } \|a-x\|_2 \leq \lambda\}$ is the λ-neighbourhood of A.

If $A, B \subset \mathbb{R}^d$ are convex polytopes, Amenta [9] shows how to simplify the calculation of the Hausdorff distance as follows: Let $\partial U_\lambda(A)$ denote the boundary of $U_\lambda(A)$. See also Figure 2.2, where $\partial U_\lambda(\Delta)$ is drawn using dashed line segments and circular arcs. Now, $\vec{D}_{\mathcal{H}}(A,B) \leq \lambda$ if every vertex of A lies within $\partial U_\lambda(B)$. This implies that $D_{\mathcal{H}}(A,B) \leq \lambda$ if every vertex of one set lies within the λ-neighbourhood of the other set. Amenta further shows that a translation minimizing the Hausdorff distance between two convex polytopes A and B can be found in linear time.

We prove a basic but fundamental result which enables us to use the abstract approximation algorithms introduced in Section 1.5.

Theorem 2.1. *For any compact subset $A \in \mathcal{C}^d$ and any translation $\tau \in \mathbb{R}^d$ we have*

$$D_{\mathcal{H}}(A, \tau(A)) \;=\; \|\tau\|_2.$$

Proof. Find a point $a \in A$ minimizing $\langle \tau, x\rangle$ over all $x \in \mathbb{R}^d$, where $\langle x, y\rangle$ denotes the standard scalar product in Euclidean space. This point has to lie in the neighbourhood of some point in $\tau(A)$. Of course, any translated point has a distance of at least $\|\tau\|_2$ to this point. Thus $D_{\mathcal{H}}(A,\tau(A)) \geq \|\tau\|_2$. Conversely, since any point has a distance of $\|\tau\|_2$ to its translated version and vice versa, the theorem is proven. □

2.3 Reference Points for the Hausdorff Distance

In this section we review known $D_{\mathcal{H}}$-reference points.

2.3.1 The Lower Left Corner

Let $A \in \mathcal{C}^d$ be a compact subset of $\mathbb{R}^d$. Let $\mathrm{LL}(A)$ be the point in $\mathbb{R}^d$ where the j-th coordinate of $\mathrm{LL}(A)$ is the minimum of all j-th coordinates of all points in A. In $\mathbb{R}^2$ this describes the lower left corner of the smallest axis-parallel rectangle enclosing A. The mapping $\mathrm{LL}\colon \mathcal{C}^d \to \mathbb{R}^d$ is a reference point for $\mathcal{C}^d$ with respect to translations. Alt, Behrends and Blömer [4] show this result for compact subsets in the plane. The extension to higher dimension is analogous to a result by Efrat, Itai and Katz [25] for point sets in $\mathbb{R}^d$.

Theorem 2.2. *[4, 25] The lower left corner* $\mathrm{LL}\colon \mathcal{C}^d \to \mathbb{R}^d$ *is a* $D_{\mathcal{H}}$*-reference point with respect to translations. Its Lipschitz constant is* $\sqrt{d}$*.*

The following theorem for sets in the plane is also obtained by Alt, Behrends and Blömer [4]. Again it can be extended to higher dimensions using the result by Efrat et al. [25]. Its proof is easy using Theorem 2.1 and the abstract algorithm for translations, see Theorem 1.5.

Theorem 2.3. *[4, 25] The* $D_{\mathcal{H}}$*-reference point* $\mathrm{LL}\colon \mathcal{C}^d \to \mathbb{R}^d$ *induces an approximation algorithm for the Hausdorff distance under translations with approximation factor* $1 + \sqrt{d}$*. The runtime of this algorithm is the time to compute the Hausdorff distance between compact sets in* $\mathbb{R}^d$*.*

In fact, Efrat, Itai and Katz [25] prove that the result can also be extended to the Hausdorff distance defined on arbitrary L_p-norm on the underlying space $\mathbb{R}^d$. The Lipschitz constant in this case is $\sqrt[p]{d}$ for $1 \leq p < \infty$ and 2 for $p = \infty$. The approximation algorithm carries over.

Theorem 2.2 can be generalized to every fixed corner of the smallest axis-parallel hyperrectangle enclosing a subset of $\mathbb{R}^d$. By Theorem 1.2 we see that every convex combination of those corners is a $D_{\mathcal{H}}$-reference point with respect to translations. This especially holds for the center which might lead to better approximations in practical applications, see also Section 6.4 for the bottleneck distance.

2.3.2 The Center of Mass of the Boundary of the Convex Hull

Alt, Behrends and Blömer [4] show that the center of mass of the boundary of the convex hull of a compact subset in the plane is a reference point for the Hausdorff distance with respect to similarities. This is the first mapping shown to be a reference point with respect to this class of transformations.

Theorem 2.4. *[4] The center of mass of the boundary of the convex hull of a compact convex subset in the plane is a* $D_{\mathcal{H}}$*-reference point with respect to similarities. Its Lipschitz constant is at most* $4\pi + 3$*.*

Again, we get an approximation algorithm for the Hausdorff distance under translations in the plane by Theorem 2.1 and the abstract approximation algorithm, see Theorem 1.5.

Theorem 2.5. *[4] In the plane, the center of mass of the boundary of the convex hull as a* $D_{\mathcal{H}}$*-reference point induces an approximation algorithm for the Hausdorff distance under*

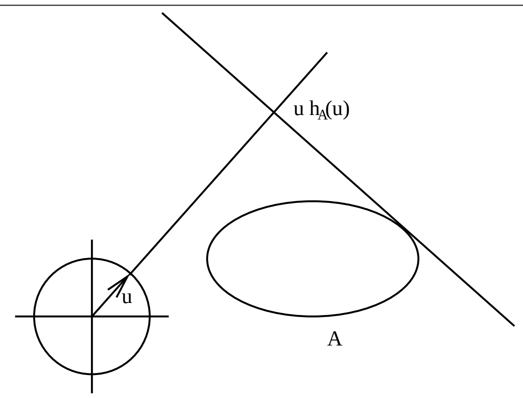


Figure 2.3: Integration of $u\,h_A(u)$ over all points $u \in \mathbb{S}^{d-1}$ leads to the Steiner point.

translations with approximation factor $4\pi + 4$. The runtime of this algorithm is the time to compute the Hausdorff distance in the plane.

2.3.3 The Steiner Point

Alt, Aichholzer and Rote [3] introduce the Steiner point as a $D_{\mathcal{H}}$-reference point with respect to similarities in any dimension. This point is also known as the Steiner curvature point or curvature centroid and has been intensively studied in the field of convex geometry, see Grünbaum [31], Shepard [47], and Schneider [46].

Definition 2.2 (Steiner point)**.** [3] Let $\mathcal{B}^d$ be the d-dimensional unit ball and $\mathbb{S}^{d-1}$ its boundary. Let $A \in \mathcal{C}^d$ be a compact subset of $\mathbb{R}^d$. Then the support function $h_A \colon \mathbb{R}^d \to \mathbb{R}$ of A is given by

$$h_A(u) \;:=\; \max_{a \in A} \langle a, u \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the standard scalar product in $\mathbb{R}^d$. The Steiner point of A is defined as

$$S(A) \;:=\; \frac{d}{\mathrm{Vol}(\mathbb{S}^{d-1})} \int_{\mathbb{S}^{d-1}} u\, h_A(u) d\omega(u),$$

see Figure 2.3 for an illustration of the integrand.

The following theorem is well-known, see Grünbaum [31] and Shepard [47].

Theorem 2.6. *[31, 47] The Steiner point of a convex polytope is the weighted sum of its vertices, where the weight of a vertex v is that fraction of the surface of the unit sphere that lies between the unit vectors normal to the hyperplanes meeting at v (the normalized exterior angle at v).*

Alt, Aichholzer and Rote [3] show that the Steiner point is a $D_{\mathcal{H}}$-reference point. Let Γ denote the Gamma function.

Theorem 2.7. *[3] The Steiner point is a $D_{\mathcal{H}}$-reference point with respect to similarities in any dimension d. Its Lipschitz constant is $\chi_d = 2\Gamma(d/2+1)/(\sqrt{\pi}\cdot\Gamma(d/2+1/2))$, which for $d = 2$ is $4/\pi$, for $d = 3$ is $3/2$ and for arbitrary dimension d lies between $\sqrt{2/\pi}\sqrt{d}$ and $\sqrt{2/\pi}\sqrt{d+1}$.* □

Alt, Aichholzer and Rote [3] state the following approximation algorithm for the Hausdorff distance under translations. Again, the proof is easy using Theorem 2.1 and the abstract approximation algorithm, see Theorem 1.5.

Theorem 2.8. *[3] The Steiner point as a $D_{\mathcal{H}}$-reference point induces an approximation algorithm for the Hausdorff distance under translations with approximation ratio $1+\chi_d$. The runtime of this algorithm is the time to compute the Steiner points plus the time to compute the Hausdorff distance.*

Alt, Aichholzer and Rote [3] give two sets proving the lower bound of $4/\pi$ for the Lipschitz constant of the Steiner point as a $D_{\mathcal{H}}$-reference point in the plane. We recall the two sets in Section 2.3.5. First we show a way how to compute the Steiner point in the plane.

2.3.4 Steiner Point in the Plane

Let $A \in \mathcal{C}^2$ be any compact subset of $\mathbb{R}^2$. Let $S_x(A)$, $S_y(A)$ be the x- and y-coordinate of the Steiner point of A, respectively. Then both values can be computed independently according to the following formulas:

$$\begin{aligned} S(A) &= \frac{1}{\pi}\int_{S^1} h_A(u)\, u\, d\omega(u) \\ &= \frac{1}{\pi}\int_0^{2\pi} h_A((\cos t, \sin t)^T)\cdot(\cos t, \sin t)^T dt \end{aligned}$$

which implies

$$\begin{aligned} S_x(A) &= \frac{1}{\pi}\int_0^{2\pi} h_A((\cos t, \sin t)^T)\cdot \cos t\, dt \\ \text{and}\quad S_y(A) &= \frac{1}{\pi}\int_0^{2\pi} h_A((\cos t, \sin t)^T)\cdot \sin t\, dt. \end{aligned}$$

In the following example we show that the Steiner point of a circle is its center. According to the equivariance under translations, it is enough to show this for circles centered at the origin.

Example. Let C be the circle in $\mathbb{R}^2$ centered at the origin with radius R. Then,

$$\begin{aligned} S_x(C) &= \frac{1}{\pi}\int_0^{2\pi} h_C((\cos t, \sin t)^T)\cdot \cos t\, dt \\ &= \frac{1}{\pi}\int_0^{2\pi} R\cdot \cos t\, dt \\ &= \frac{R}{\pi}(\sin(2\pi) - \sin(0)) = 0, \end{aligned}$$

since $h_C((\cos t, \sin t)^T) \equiv R$. An analogous calculation shows $S_y(C) = 0$.

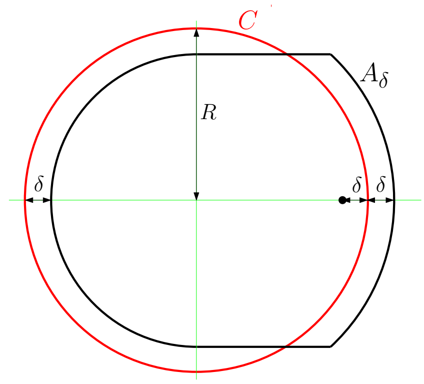


Figure 2.4: Non-convex lower bound C, A_δ given by Alt et al. [3].

2.3.5 Non-Convex Lower Bound for the Lipschitz Constant of the Steiner Point in the Plane

Alt, Aichholzer and Rote [3] give two sets $C, A_\delta \in \mathcal{C}^2$ to prove the lower bound of $4/\pi$ for the Lipschitz constant of the Steiner point. The set C is the circle of radius R around the origin. The set A_δ consists of a "distorted" circle and an additional point. See Figure 2.4 for an illustration.

2.3.6 Convex Lower Bound for the Lipschitz Constant of the Steiner Point in the Plane

One of the sets proving the lower bound for the Lipschitz constant of the Steiner point given by Alt, Aichholzer and Rote [3] is non-convex. We slightly modify this set A_δ and thereby construct a new set $B_\delta \in \mathcal{C}^2$, such that the circle C with radius R around the origin together with this set proves the lower bound, and both sets are convex. See Figure 2.5 for an illustration of C and B_δ. The modified set can be constructed as follows: We add an additional point at coordinates $(-R-\delta, 0)$ to A_δ and take the convex hull of this set. It is easy to check that C and B_δ are illustrated in optimal position with respect to translations and $D_{\mathcal{H}}^{\text{opt}}(C, B_\delta) = D_{\mathcal{H}}(C, B_\delta) = \delta$.

We now show that the x-coordinate of the Steiner point $S_x(B_\delta)$ tends to $4\delta/\pi$ as the radius R tends to infinity. See Figure 2.6 for the notation in the following calculation.

Let $f(t)$ denote the support function for points u of the unit circle with an angle $\alpha \le t \le \pi/2$ to the x-axis. The trace of $u \cdot h_{B_\delta}(u) = u \cdot f(t)$ is represented by the dashed arc in Figure 2.6. Analogously, $g(t)$ denotes the support function of B_δ for points with an angle between γ and π.

$$\begin{aligned} S_x(B_\delta) &= \frac{1}{\pi}\int_0^{2\pi} h_A((\cos t, \sin t)^T) \cdot \cos t \, dt \\ &= \frac{2}{\pi}\left(\int_0^{\alpha} (R+\delta)\cdot \cos t\, dt + \int_{\alpha}^{\pi/2} f(t)\cdot \cos t\, dt \right.\\ &\quad \left. + \int_{\pi/2}^{\gamma} (R-\delta)\cdot \cos t\, dt + \int_{\gamma}^{\pi} g(t) \cos t\, dt\right) \end{aligned}$$

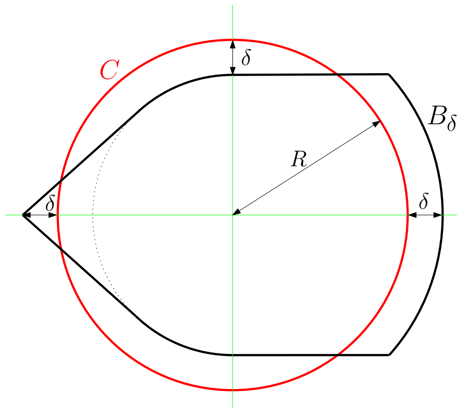


Figure 2.5: Convex lower bound.

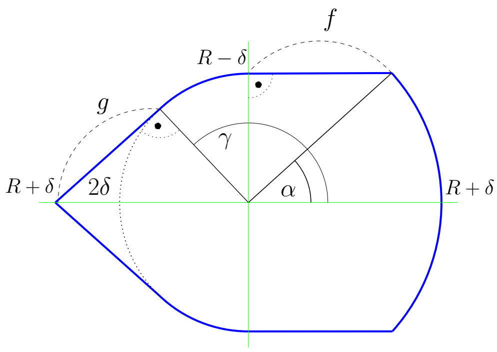


Figure 2.6: Notation for the proof of the convex lower bound.

Since $f(t) \geq R - \delta$ and $g(t) \leq R + \delta$ we have

$$\begin{aligned} S_x(B_\delta) &\geq \frac{2}{\pi}\Bigg((R+\delta)\cdot\int_0^\alpha \cos t\,dt + (R-\delta)\cdot\int_\alpha^{\pi/2}\cos t\,dt \\ &\qquad + (R-\delta)\cdot\int_{\pi/2}^{\gamma}\cos t\,dt + (R+\delta)\cdot\int_\gamma^\pi \cos t\,dt\Bigg) \\ &= \frac{4\delta}{\pi}(\sin\alpha - \sin\gamma). \end{aligned}$$

Now,

$$\sin\alpha = \cos(\pi/2 - \alpha) = (R-\delta)/(R+\delta),$$

which tends to 1 as R tends to ∞, and

$$\sin\gamma = \sqrt{1-\cos^2\gamma} = \sqrt{1-\cos^2(\pi-\gamma)} = \sqrt{1-\frac{(R-\delta)^2}{(R+\delta)^2}},$$

which tends to 0 as R tends to ∞. Therefore,

$$\lim_{R\to\infty} S_x(B_\delta) \geq 4\delta/\pi.$$

Further, by the upper bound on the Lipschitz constant we have

$$\|S(B_\delta)\|_2 = \|S(B_\delta) - S(C)\|_2 \leq \frac{4}{\pi} D_{\mathcal{H}}(B_\delta, C) = \frac{4\delta}{\pi},$$

implying that $S_x(B_\delta) \leq 4\delta/\pi$, thus

$$\lim_{R\to\infty} S_x(B_\delta) = 4\delta/\pi.$$

Similarly we can compute $S_y(B_\delta) = 0$. Another way to see this is by observing that B_δ has the x-axis as reflection line.

Again, we combine this result with the fact that the Steiner point of C is the origin, see Example 2.3.4, and the placement of the two sets in the above figure leads to a Hausdorff distance of δ. Thereby we see that the Lipschitz constant in this example tends to $4/\pi$ as R tends to ∞.

2.4 Rigid Motions and Similarities

Alt, Aichholzer and Rote [3] use the abstract approximation algorithms given in Sections 1.5.3 and 1.5.4 to determine approximation algorithms for rigid motions and similarities in the plane and 3-space. They use the exact algorithms by Alt, Behrends and Blömer [4] to find an optimal rotation around a fixed point. In the plane this can be done in $O(nm \log nm \log^* nm)$ time using Davenport-Schinzel sequences. In 3-space the solution can be found in $O((nm)^3 \cdot T^{D_{\mathcal{H}}}(n,m))$ time, where $T^{D_{\mathcal{H}}}(n,m)$ denotes the time to compute the Hausdorff distance in $\mathbb{R}^3$.

For similarities they use the approximate scaling given by the quotient of the diameters of the two sets. This leads to the increased approximation ratio in this case. Summarizing, they prove the following theorem:

Theorem 2.9. *[3] Let $r\colon \mathcal{C}^d \to \mathbb{R}^d$ be a $D_{\mathcal{H}}$-reference point with respect to rigid motions or similarities and with Lipschitz constant c, where $d = 2$ or 3. We can find an approximately optimal matching for rigid motions or similarities with approximation factor $c+1$ or $c+3$, respectively. The runtime in $\mathbb{R}^2$ is $O(nm \log nm \log^* nm)$, and in $\mathbb{R}^3$ it is $O((nm)^3 \cdot T^{D_{\mathcal{H}}}(n,m))$.*

2.5 General Lower Bound for the Lipschitz Constant in the Plane

In this section we investigate lower bounds for the Lipschitz constant of an arbitrary $D_{\mathcal{H}}$-reference point with respect to translations in the plane.

Alt, Aichholzer and Rote [3] prove that the Steiner point is a reference point with Lipschitz constant $4/\pi$. Using results by Rutovitz [45], Daugavet [23], and Przesławski and Yost [42], they prove that the Lipschitz constant of any $D_{\mathcal{H}}$-reference point with respect to translations from $\mathcal{C}^2$ into $\mathbb{R}^2$ is at least $4/\pi$. Their proof however is based on the Hahn-Banach Theorem, which is based on (a weaker version of) the Axiom of Choice. Consequently, the proof is non-constructive. This lower bound even holds in the case when restricted to the set of all convex subsets $\mathcal{K}^2$. Thus it is interesting to find (convex) subsets of $\mathbb{R}^2$ proving lower bounds for the Lipschitz constant of a $D_{\mathcal{H}}$-reference point. In Section 1.4.4 we prove a lower bound of 1 for the Lipschitz constant of any reference point with respect to translations for arbitrary distance measures. For this we use only convex sets.

Alt, Aichholzer and Rote [3] give three sets in the plane proving that the Lipschitz constant of any $D_{\mathcal{H}}$-reference point is greater than or equal to $\sqrt{4/3} \approx 1.155$. This does not quite match the upper bound of $4/\pi \approx 1.273$. One of the sets used to prove this bound is non-convex, whereas the lower bound based on the results by Przesławski and Yost [42] holds even when restricted to convex sets. Alt, Aichholzer and Rote [3] pose the question to find better constructions that either give a better bound or use convex sets only. We address the second part and show three convex shapes which prove the same lower bound. The technique used equals the one of Alt et al. [3].

Let C denote the circle with radius 2 around the origin. Let Δ be the equilateral triangle with side length 4 and vertices q_1, q_2 and q_3 in counterclockwise order. Let q_1 be the origin. See Figure 2.7 for an illustration. These two sets equal the sets used in the lower bound by Alt, Aichholzer and Rote [3]. We substitute the non-convex set from this proof by the third set which is depicted in Figure 2.7 together with Δ and C. We denote this set by K. This set is the convex hull of the circle with radius 1 around the origin, and the intersection points of the circles around q_2 and q_3 with radius 1 and the non-horizontal sides of Δ.

Theorem 2.10. *Let $r\colon \mathcal{K}^2 \to \mathbb{R}^2$ be any $D_{\mathcal{H}}$-reference point with respect to translations and with Lipschitz constant c. Then, $c \geq \sqrt{4/3}$.*

Proof. Let us first assume that the reference point $r(C)$ of the circle C is its center, thus equals the origin. See Figure 2.7 for an illustration of the following.

Observe that $D_{\mathcal{H}}(C, K) = 1$. Since $\|r(K) - r(C)\|_2 \leq c \cdot D_{\mathcal{H}}(C, K)$ it follows that $r(K)$ has to lie in a circle of radius c around the origin. Similarly, $D_{\mathcal{H}}(\Delta, K) = 1$ and therefore $r(\Delta)$ has to lie in a circle of radius c around $r(K)$ and thus in a circle of radius $2c$ around the origin. Since q_1 equals the origin, $r(\Delta)$ has to lie in a circle of radius $2c$ around q_1. Now we rotate K by angles of $2\pi/3$ and $4\pi/3$ counterclockwise around the origin and translate Δ such that q_2 and q_3, respectively, are placed in the origin. By the same considerations as above and using the equivariance under translations, we see that the reference point of Δ also has to lie in a circle of radius $2c$ around q_2 and q_3. Thus, the three circles around the corners have to intersect and therefore, $2c$ has to be greater than or equal to $2/3$ times the height of Δ, that is $2\sqrt{12}/3$. This implies $c \geq \sqrt{4/3}$.

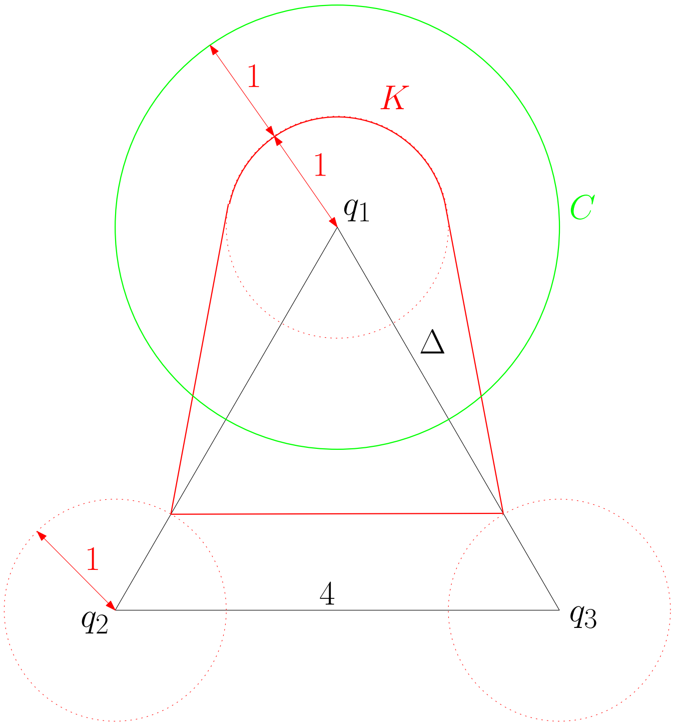


Figure 2.7: The convex sets Δ, K and C.

If the reference point $r(C)$ of the circle C is not its center, the only difference is that the centers of the final three circles around the vertices of Δ are translated by the vector from the center to $r(C)$. □

2.6 Lower Bounds for the Approximation Algorithm in the Plane

It is interesting to determine sets of shapes where the approximation algorithm leads to bad results, no matter which reference point we use. That is, we want to find sets proving a lower bound for the approximation algorithm for translations using reference points. We concentrate on convex sets in the plane.

Review the reference point method: Given a reference point with respect to translations and with Lipschitz constant c, and a set of shapes $A_1, \ldots, A_n$, we have that

$$D_{\mathcal{H}}(A_i - r(A_i), A_j - r(A_j)) \;\leq\; (1+c) \cdot D_{\mathcal{H}}^{\mathrm{opt}}(A_i, A_j)$$

for all $1 \leq i < j \leq n$.

Hence, to find a lower bound γ for the approximation factor of the algorithm, we have to find a set of shapes $A_1, \ldots, A_n$, such that for any reference point r there exists a pair (i, j), such that

$$D_{\mathcal{H}}(A_i - r(A_i), A_j - r(A_j)) \;\geq\; \gamma \cdot D_{\mathcal{H}}^{\mathrm{opt}}(A_i, A_j).$$

By the optimality of the right side we have that $\gamma \geq 1$. The Steiner point as a $D_{\mathcal{H}}$-reference point with Lipschitz constant $4/\pi$ implies that $\gamma \leq 1 + 4/\pi$.

More general, we want to find a set of shapes $A_1, \ldots, A_n$, such that for any set of vectors $\tau_1, \ldots, \tau_n \in \mathbb{R}^2$ there exists a pair (i, j), such that

$$D_{\mathcal{H}}(A_i - \tau_i, A_j - \tau_j) \;\geq\; \gamma \cdot D_{\mathcal{H}}^{\text{opt}}(A_i, A_j),$$

where $1 \leq \gamma \leq 1 + 4/\pi$. Of course, a lower bound for this problem is also a lower bound for the reference point problem.

We have implemented a computer program to construct convex sets proving a lower bound for the latter problem. Schematically, this program works as follows: Let $A_1, \ldots, A_n$ be a given set of convex polytopes in the plane. We compute the optimal Hausdorff distance under translations for each pair (i, j) by using Lemma 2.1 and the method proposed by Amenta [9]. This step gives us the numbers $D_{\mathcal{H}}^{\text{opt}}(A_i, A_j)$ for every pair (i, j).

The second step is to find the current lower bound. By binary search we compute the minimal number κ such that the system of inequalities

$$\forall\, 1 \leq i < j \leq n \;:\; D_{\mathcal{H}}(A_i - \tau_i, A_j - \tau_j) \;\leq\; \kappa \cdot D_{\mathcal{H}}^{\text{opt}}(A_i, A_j)$$

has a solution. We approximate the circular parts of the boundary of the neighbourhood by polygons with a fixed number of vertices and then use linear programming to find the solution. Again we use ideas by Amenta [9] to solve this problem.

In a third step, we use the dual variables of this solution to construct new convex sets which, by adding to the current solution, lead to a higher lower bound. We eliminate figures which do not contribute at the current state.

One of the best results we got by this program is the set of 53 convex sets depicted in Figure 2.8. The sets in this figure are drawn in optimal position with respect to translations and lead to an approximation ratio of approximately 1.63. Removing an arbitrary set decreases the induced lower bound. The points in the center are the Steiner points of the sets.

Later on we generated sets which lead to an approximation ratio slightly higher than 2. These sets look similar to the one shown in Figure 2.8. Unfortunately, we cannot theoretically prove the lower bound on the approximation ratio. The incremental construction by the program leads to sets which are very similar and thus, rounding and approximation errors in the computation are involved for sure.

We were able to extract the following three convex sets from this set of shapes, see Figures 2.10 and 2.9.

Let Δ be the equilateral triangle with side length 4. Let q_1, q_2 and q_3 denote the vertices of Δ in counterclockwise order. Let $h_\Delta := 2\sqrt{3}$ denote the height of Δ. Let Δ^R be Δ rotated by π counterclockwise around the center of mass $C(\Delta)$ of Δ. Let q_1^R, q_2^R and q_3^R denote the rotated vertices.

We construct the third set as follows: Consider the rectangle with length h_Δ and width 2. Place a half circle of radius 1 at both short sides of the rectangle. We denote the convex hull of the rectangle and the two halfcircles by D. Let $h_D := (2 + 2\sqrt{3})$ denote the height of D. See Figure 2.10 for an illustration of Δ and D. Obviously, these two sets are in unique optimal position with respect to translations.

We use the following lemma to prove the main result of this section.

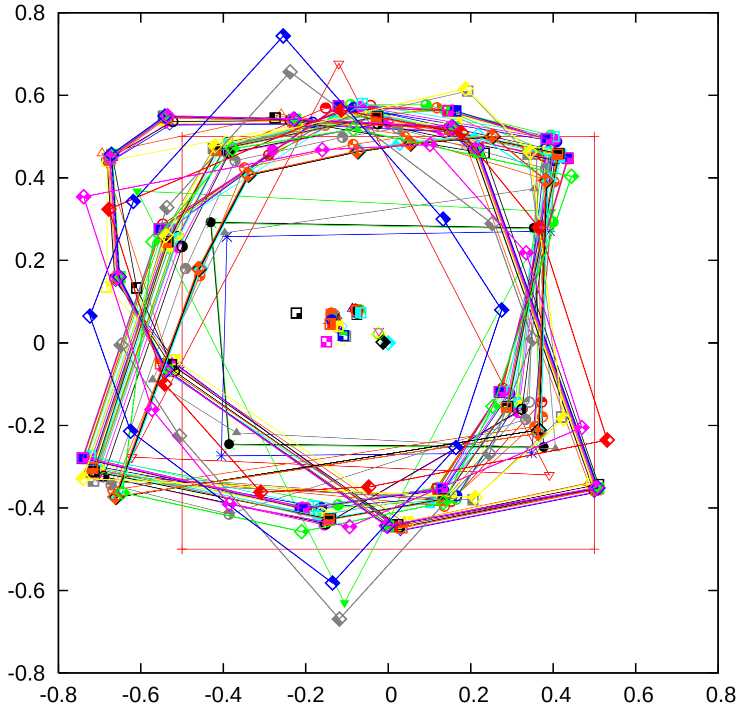


Figure 2.8: Convex sets generated by the computer program.

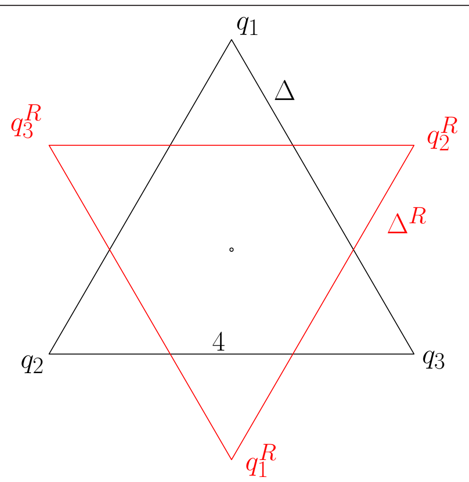


Figure 2.9: Construction of Δ and Δ^R.

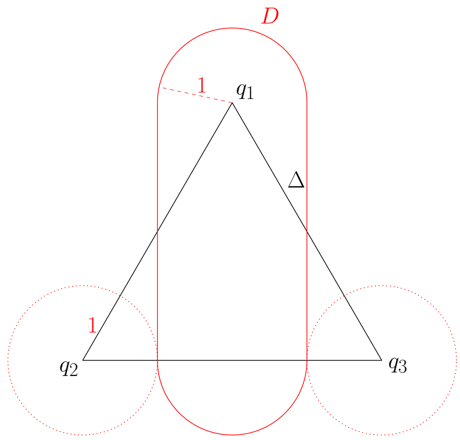


Figure 2.10: Construction of Δ and D.

Lemma 2.2. *Let $\tau^{\Delta}, \tau^{\Delta^R}, \tau^D \in \mathbb{R}^2$ be transalation vectors. Let $\gamma^* := 1 + \sqrt{1/3} \approx 1.58$. Then,*

$$\begin{array}{rrcl} \textit{either} & D_{\mathcal{H}}(\Delta - \tau^{\Delta}, D - \tau^D) & \geq & \delta \\ \textit{or} & D_{\mathcal{H}}(\Delta^R - \tau^{\Delta^R}, D - \tau^D) & \geq & \delta, \end{array}$$

where $\delta = \gamma^* + (C(\Delta)_y - C(\Delta^R)_y)/2$.

Proof. With a slight abuse of notation, let $\Delta := \Delta - \tau^{\Delta}$, $\Delta^R := \Delta^R - \tau^{\Delta^R}$ and $D := D - \tau^D$. Let a and b be the top and bottom point of D, respectively. See Figure 2.11 for the construction.

$$\begin{aligned} & D_{\mathcal{H}}(\Delta^R, D) + D_{\mathcal{H}}(\Delta, D) \\ \geq \quad & a_y - (q_3^R)_y + (q_3)_y - b_y \\ = \quad & a_y - (q_3^R)_y + (q_3)_y - (a_y - h_D) \\ = \quad & -(q_3^R)_y + (q_3)_y + h_D \\ = \quad & h_D - 2/3\, h_{\Delta} - C(\Delta^R)_y + C(\Delta)_y, \quad \text{see Figure 2.11} \\ = \quad & 2 + 2\sqrt{3} - 4/3\,\sqrt{3} - C(\Delta^R)_y + C(\Delta)_y \\ = \quad & 2 + 2/3\,\sqrt{3} - C(\Delta^R)_y + C(\Delta)_y \end{aligned}$$

This implies that $\max\{D_{\mathcal{H}}(\Delta^R, D), D_{\mathcal{H}}(\Delta, D)\} \geq 1 + 1/3\,\sqrt{3} + 1/2\,(C(\Delta)_y - C(\Delta^R)_y)$

□

We now use this lemma and the invariance of Δ and Δ^R under rotations to prove the lower bound on the approximation ratio of our approximation algorithm using reference points.

Theorem 2.11. *Let $r\colon \mathcal{K}^2 \to \mathbb{R}^2$ be a $D_{\mathcal{H}}$-reference point with respect to translations. The approximation ratio γ of the induced approximation algorithm for the Hausdorff distance under translations is at least $\gamma^* := 1 + \sqrt{1/3} \approx 1.58$.*

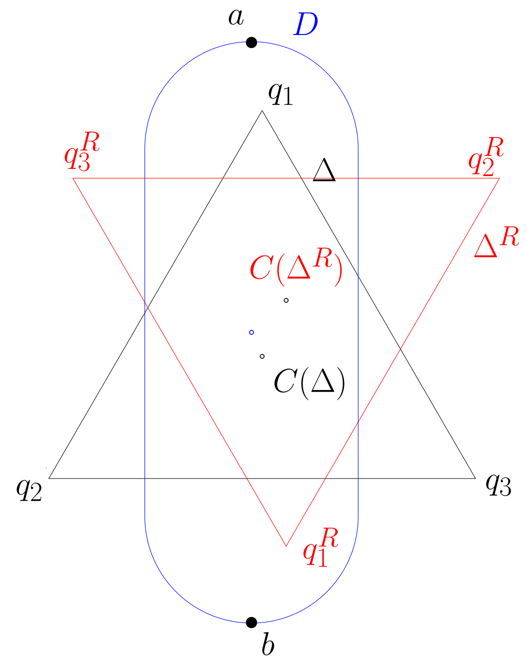


Figure 2.11: Translated versions of Δ, Δ^R and D.

Proof. Let $D^1 := D$, let D^2 be D rotated by $2\pi/3$ and D^3 be D rotated by $4\pi/3$ as shown in Figure 2.12. It is easy to see that $D_{\mathcal{H}}^{\text{opt}}(D^i, \Delta) = 1 = D_{\mathcal{H}}^{\text{opt}}(D^i, \Delta^R)$. Let the set $\mathcal{X} := \{D^1, D^2, D^3, \Delta, \Delta^R\}$ be positioned in the plane such that

$$D_{\mathcal{H}}(A, B) \;\leq\; \gamma \cdot D_{\mathcal{H}}^{\text{opt}}(A, B)$$

for all $A, B \in \mathcal{X}$ with a minimal $1 \leq \gamma \leq 1 + 4/\pi$.

If the centers of mass $C(\Delta)$ and $C(\Delta^R)$ coincide, then $\gamma = \gamma^*$ by Lemma 2.2 applied to Δ, Δ^R and D^1. Let ℓ_1 be a vertical line oriented from bottom to top. Let ℓ_2 and ℓ_3 be ℓ_1 rotated by $2\pi/3$ and $4\pi/3$, respectively. There is at least one index $i^* \in \{1, 2, 3\}$, such that the directed distance $\text{Proj}_{i^*}(C(\Delta)) - \text{Proj}_{i^*}(C(\Delta^R)) > 0$, where Proj_i denotes the projection on the line ℓ_i. Then by Lemma 2.2 either $D_{\mathcal{H}}(\Delta, D^{i^*}) > \gamma^*$ or $D_{\mathcal{H}}(\Delta^R, D^{i^*}) > \gamma^*$ for any position of D^{i^*}. □

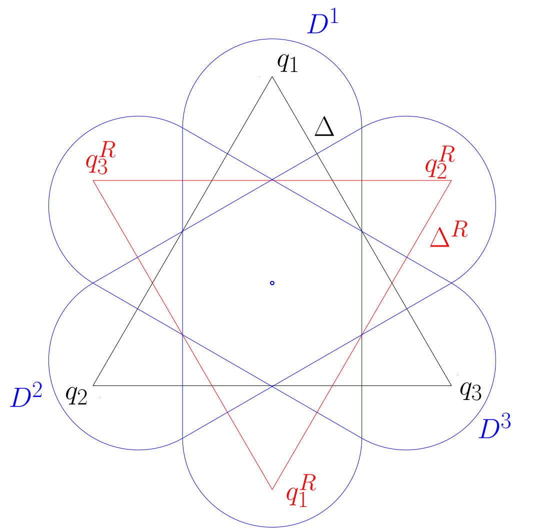


Figure 2.12: The set $\mathcal{X}$ of shapes proving the lower bound of $1 + \sqrt{1/3}$.

Chapter 3

Earth Mover's Distance

The Earth Mover's Distance (EMD) is a useful distance measure on weighted point sets with applications in shape matching, color-based image retrieval and music score matching. See the work of Assent, Wenning and Seidl [10], Cohen [19], Cohen and Guibas [21], Giannopoulos and Veltkamp [27], Graumann and Darell [30], Typke, Giannopoulos, Veltkamp, Wiering and Oostrum [48], and Rubner, Tomasi and Guibas [44] for more information. For these applications it is useful to have a quick estimation on the minimum distance between two weighted point sets which can be achieved under a given class of transformations. For the EMD this problem was first regarded by Cohen [19] and Cohen and Guibas [21]. They constructed an iterative Flow-Transformation algorithm, which they proved to converge, but not necessarily to the global minimum. In this chapter we show that we can apply our reference point framework to obtain fast constant-factor approximations on the EMD under translations, rigid motions and similarities. Recently, Cabello, Giannopoulos, Knauer and Rote [15] considered similar problems. The advantage of our approach is that the results can be applied to arbitrary dimension and any norm on the underlying space. Therefore the results are widely applicable.

Parts of this chapter have been published by Klein and Veltkamp [37, 38].

3.1 Results

For weighted point sets in the plane, we show a 2-approximation algorithm for translations, a 4-approximation algorithm for rigid motions and an 8-approximation algorithm for similarities. The runtime of the approximation for translations is $O(T^{\text{EMD}}(n,m))$, the runtime of the other two algorithms is $O(nm \cdot T^{\text{EMD}}(n,m))$, where $T^{\text{EMD}}(n,m)$ denotes the time to compute the EMD between two weighted point sets with n and m points, respectively. We also show that these algorithms can be extended to arbitrary dimension, leading however to worse time and approximation bounds. All these algorithms are based on EMD-reference points, which allows the elegant generalizations to higher dimensions. We give a comprehensive discussion of EMD-reference points for weighted point sets.

In Chapter 4 we present more results concerning the EMD. There we prove that the time to compute the EMD based on the L_1-norm exactly can be reduced significantly to $O(n^2 \log^{2d-1} n)$ by using small Manhattan spanners, where $d \geq 2$ denotes the dimension.

3.2 Basic Definitions

Rubner, Tomasi and Guibas [44] investigate the application of the EMD for image retrieval and shape matching. In their work images are described as histograms or signatures, i.e., weighted point sets:

Definition 3.1 (Weighted Point Set)**.** [27, 44] Let $p_1, \ldots, p_n \in \mathbb{R}^d$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{R}_{>0}$. Then, a pair $a_i = (p_i, \alpha_i)$ for $i = 1, \ldots, n$ is called a weighted point in $\mathbb{R}^d$ and the set $A = \{a_1, \ldots, a_n\}$ is called a weighted point set. We call α_i the weight of p_i and $W^A = \sum_{i=1}^{n} \alpha_i$ the total weight of A. We write $\mathbb{W}^d$ for the set of all weighted point sets in $\mathbb{R}^d$ and $\mathbb{W}^{d,G}$ for the set of all weighted point sets in $\mathbb{R}^d$ with total weight $G \in \mathbb{R}_{>0}$.

In the following we use the considered class of transformations on both weighted point sets and discrete subsets of $\mathbb{R}^d$. By a transformation on a weighted point set we mean that we transform the coordinates of the weighted points and leave their weights unchanged.

We now introduce the center of mass, a point associated to each weighted point set. This point plays an important role in our approximation algorithms. The computation time of this point is linear, and therefore it does not affect the runtime of any of our algorithms.

Definition 3.2 (Center of Mass)**.** Let $A = \{(p_i, \alpha_i)_{i=1,\ldots,n}\} \in \mathbb{W}^{d,G}$ be a weighted point set for some $G \in \mathbb{R}_{>0}$. The center of mass of A is defined as

$$C(A) \;=\; \frac{1}{W^A} \sum_{i=1}^{n} \alpha_i p_i .$$

As we will see later, the center of mass is an EMD-reference point. Next, we define the EMD for arbitrary distance measures on the underlying space $\mathbb{R}^d$, though we restrict all later considerations to the case where this distance measure is a norm.

Definition 3.3 (Earth Mover's Distance)**.** [19] Let $A = \{(p_i, \alpha_i)_{i=1,\ldots,n}\}$, $B = \{(q_j, \beta_j)_{j=1,\ldots,m}\}$ in $\mathbb{W}^d$ be two weighted point sets with total weights W^A, $W^B > 0$. Let $h\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be the underlying distance measure on the underlying space $\mathbb{R}^d$. The Earth Mover's Distance between A and B is defined as

$$\mathrm{EMD}(A, B) \;=\; \frac{\min_{F \in \mathcal{F}} \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}\, h(p_i, q_j)}{\min\{W^A, W^B\}},$$

where the minimum is taken over the set $\mathcal{F}$ of feasible flows $F = \{f_{ij}\}$.

$$\begin{aligned}
f_{ij} &\;\geq\; 0, && \text{for } i = 1, \ldots, n,\ j = 1, \ldots, m \\
\sum_{j=1}^{m} f_{ij} &\;\leq\; \alpha_i, && \text{for } i = 1, \ldots, n \\
\sum_{i=1}^{n} f_{ij} &\;\leq\; \beta_j, && \text{for } j = 1, \ldots, m \\
\sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij} &\;=\; \min\{W^A, W^B\}.
\end{aligned}$$

Throughout this chapter we concentrate on weighted point sets with equal total weight. In this case, the calculation of the EMD can be simplified as stated in the following Lemma 3.1. The proof of this lemma follows immediately by the non-negativity of the flow variables.

Lemma 3.1. *Let $A = \{(p_i, \alpha_i)_{i=1,\dots,n}\}$ and $B = \{(q_j, \beta_j)_{j=1,\dots,m}\} \in \mathbb{W}^{d,G}$ be weighted point sets with equal total weight $G \in \mathbb{R}_{>0}$. Let $h\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be a distance measure on the underlying space $\mathbb{R}^d$. Then the Earth Mover's Distance between A and B equals*

$$\mathrm{EMD}(A, B) \;=\; \frac{1}{G} \cdot \min_{F \in \mathcal{F}} \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}\, h(p_i, q_j),$$

where $F = \{f_{ij}\}$ is a feasible flow, i.e.,

$$\begin{aligned} f_{ij} &\geq 0, && \text{for } i = 1, \dots, n,\ j = 1, \dots, m \\ \sum_{j=1}^{m} f_{ij} &= \alpha_i, && \text{for } i = 1, \dots, n \\ \sum_{i=1}^{n} f_{ij} &= \beta_j, && \text{for } j = 1, \dots, m. \end{aligned}$$

The following properties of the EMD have been proven by Rubner, Tomasi and Guibas [44], and Giannopoulos and Veltkamp [27]: The EMD is a metric if the underlying distance is a metric and is applied on the space of weighted point sets with equal total weight. It is continuous and insensitive against noise. The EMD allows for partial matching by definition, that is we can compute the EMD between points sets of different total weight. Unfortunately our reference point approach does not allow for partial matching, see Section 3.3.1. Drawbacks occur when the EMD is applied to non-equal total weight sets. In this case, it does not obey the positivity property, does not take into account the surplus of weight, cannot distinguish between two non-identical sets and, most important, it does not obey the triangle inequality.

An upper bound for the time to compute the EMD is $O(n^4 \log n)$ using a strongly polynomial minimum cost flow algorithm by Orlin [41]. In practice, an algorithm using the simplex method to solve the linear program is usually faster. Cabello et al. [15] give a $(1 + \varepsilon)$-approximation algorithm with runtime $O(n^2 \varepsilon^{-2} \log^2(n\varepsilon^{-1}))$. Indyk [36] proves an $O(n \log^{O(1)} n)$-time randomized $O(1)$-approximation algorithm if the two point sets consist of an equal number of unit weight points in $\mathbb{R}^2$. Using a Manhattan network as a 1-spanner for the L_1-norm, we can compute the L_1-EMD in d dimensions in $O(n^2 \log^{2d-1} n)$ time using Orlin's algorithm on the reduced graph, see Chapter 4. This improves the previously best known runtime of $O(n^4 \log n)$ significantly. Further, this approach leads to a $\sqrt{2}$-approximation with the same runtime for the important case when the EMD is based on the Euclidean distance. This algorithm is conceptually easier than the $(1 + \varepsilon)$-approximation given by Cabello et al. [15].

For the rest of the chapter we assume that the underlying distance measure is a norm and hence the EMD is a metric. We further assume that this norm is the same as the one used in the definition of the EMD-reference point. If the underlying norm is any L_p-distance, where $1 \leq p \leq \infty$, we write EMD_p to denote the Earth Mover's Distance based on this norm.

We now prove a basic but fundamental result which allows us to apply the abstract approximation algorithms given in Section 1.5.

Theorem 3.1. *Let $A = \{(p_i, \alpha_i)_{i=1,\dots,n}\} \in \mathbb{W}^{d,G}$ be a weighted point set with total weight $G \in \mathbb{R}_{>0}$, and let $\tau \in \mathbb{R}^d$ be any translation vector. Then*

$$\mathrm{EMD}(A, \tau(A)) \;=\; \|\tau\|.$$

Proof.

$$\begin{aligned}
\mathrm{EMD}(A, \tau(A)) &= \frac{1}{G} \cdot \min_{F \in \mathcal{F}} \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij} \|p_i - (p_j + \tau)\| \\
&\leq \frac{1}{G} \cdot \sum_{i=1}^{n} \alpha_i \|p_i - (p_i + \tau)\|, \quad \text{by choosing } f_{ii} = \alpha_i \text{ as a feasible flow} \\
&= \frac{1}{G} \cdot \sum_{i=1}^{n} \alpha_i \|\tau\| \\
&= \|\tau\|.
\end{aligned}$$

Further, Cohen and Guibas [21] show that the distance of the centers of mass is a lower bound for the EMD, see also the proof of Theorem 3.4. Therefore

$$\mathrm{EMD}(A, \tau(A)) \;\geq\; \|C(A) - C(\tau(A))\| \;=\; \|\tau\|$$

and the lemma is proven. □

3.3 EMD-Reference Points

In this section we discuss the existence of EMD-reference points. We start with a negative result for point sets with unequal total weight.

3.3.1 Non-Existence of Reference Points for Unequal Total Weight

Theorem 3.2. *There is no* EMD*-reference point for weighted point sets with unequal total weight with respect to all transformation sets that include the set of translations.*

Proof. Assume there is an EMD-reference point r with Lipschitz constant $c \geq 0$. Let $p, q \in \mathbb{R}^d$ be any two distinct points. Define the three weighted point sets $A := \{(p, 1)\}$, $B := \{(q, 1)\}$, and $C := A \cup B$.

Since $\mathrm{EMD}(A, C) = 0$ we see by using the Lipschitz continuity that $\|r(A) - r(C)\| = 0$, which implies $r(A) = r(C)$. For the same reason we have $r(B) = r(C)$, which implies $r(A) = r(B)$. Conversely, observing that B is A translated by $q - p$, we see that $r(B)$ is $r(A)$ translated by $q - p$ using the equivariance under translation. Since $q - p \neq 0$ it follows that $r(A) \neq r(B)$, leading to a contradiction. □

We can arbitrarily choose the points p and q in the last proof. Therefore, the result is valid for weighted point sets of arbitrary diameter. Additionally, since all weights are 1, it is independent of the ratio of the weights of the points.

Corollary 3.1. *The statement of Theorem 3.2 holds even if we only consider weighted point sets of bounded diameter or bounded ratio of weights of the points.*

Unfortunately, Theorem 3.2 has a deep impact on the usability of the reference point approach for shape matching, since it makes it impossible to use this approach for partial matching applications.

We now extend Theorem 3.2 and show that there is no weak EMD-reference point for weighted point sets with unequal total weight, see Definition 1.1.

Theorem 3.3. *There is no weak* EMD*-reference point for weighted point sets with unequal total weight with respect to all transformation sets that include the set of translations.*

Proof. Assume $r\colon \mathbb{W}^{d,G} \to \mathbb{R}^d$ is a weak reference point for some $G \in \mathbb{R}_{>0}$. Let $K \geq 1$ be a large constant, O the origin and e_1 the first unit vector in $\mathbb{R}^d$. Let $\mathrm{EMD}^{\mathrm{ref}}(A, B)$ denote the Earth Mover's Distance where the weak reference points of A and B coincide. Let further $\mathrm{EMD}^{\mathrm{opt}}(A, B)$ denote the EMD under an optimal translation. Consider the following four weighted point sets:

$$\begin{aligned} A &= \{(O, 1)\} \\ B &= \{(e_1, 1)\} \\ C_1 &= \{(O, K), (e_1, 1)\} \\ C_2 &= \{(O, 1), (e_1, K)\} \end{aligned}$$

By equivariance we know that

$$r(B) = r(A) + e_1.$$

Let $i \in \{1, 2\}$. Then,

$$\begin{aligned} \mathrm{EMD}^{\mathrm{opt}}(C_i, A) &= 0 \\ \Rightarrow \mathrm{EMD}^{\mathrm{ref}}(C_i, A) &= 0 \\ \Rightarrow \mathrm{EMD}(C_i, A + r(C_i) - r(A)) &= 0 \\ \Rightarrow \mathrm{EMD}(C_i, \{(r(C_i) - r(A), 1)\}) &= 0 \\ \Rightarrow (r(C_i) - r(A) = O &\vee r(C_i) - r(A) = e_1). \end{aligned}$$

Analogously we see that $\mathrm{EMD}^{\mathrm{opt}}(C_i, B) = 0$ implies $(r(C_i) - r(B) = O \vee r(C_i) - r(B) = e_1)$. Let us now consider the four possibilities:

1. $(r(C_i) - r(A) = O \wedge r(C_i) - r(B) = O) \Rightarrow r(B) = r(A) \rightsquigarrow$ Contradiction.
2. $(r(C_i) - r(A) = O \wedge r(C_i) - r(B) = e_1) \Rightarrow r(A) - r(B) = e_1 \rightsquigarrow$ Contradiction.
3. $(r(C_i) - r(A) = e_1 \wedge r(C_i) - r(B) = O) \Rightarrow r(B) = r(C_i)$
4. $(r(C_i) - r(A) = e_1 \wedge r(C_i) - r(B) = e_1) \Rightarrow r(B) = r(A) \rightsquigarrow$ Contradiction.

Thus we have shown that $r(B) = r(C_i)$ for $i = 1, 2$ and hence $r(C_1) = r(C_2)$.

To bound the EMD between C_1 and C_2 in optimal position, we observe that $\mathrm{EMD}^{\mathrm{opt}}(C_1, C_2)$ is smaller than or equal to the EMD between those sets when C_2 is translated, such that the two points with weight K coincide. Since the underlying distance is a norm, we will always assign a non-zero flow between points of distance zero. Therefore we have

$$\mathrm{EMD}^{\mathrm{opt}}(C_1, C_2) \leq \frac{2\|e_1\|}{K+1}.$$

On the other hand, the EMD between C_1 and C_2 when the two weak reference points coincide can easily be computed to

$$\mathrm{EMD}^{\mathrm{ref}}(C_1, C_2) = \mathrm{EMD}(C_1, C_2) = \frac{(K-1)\|e_1\|}{K+1}.$$

Thus it follows

$$\frac{\mathrm{EMD}^{\mathrm{ref}}(C_1, C_2)}{\mathrm{EMD}^{\mathrm{opt}}(C_1, C_2)} \geq \frac{K-1}{2}.$$

This is a contradiction because matching with respect to the weak EMD-reference point r has to induce a constant-factor approximation, but K can be chosen arbitrarily large. □

3.3.2 The Center of Mass as an EMD-Reference Point

In the following section we present approximation algorithms for the EMD under transformations using reference points. Since this is only useful if there is an EMD-reference point, we restrain the consideration to weighted point sets with equal total weight. We show that in this case the center of mass is a reference point.

Theorem 3.4. *The center of mass is an* EMD*-reference point for weighted point sets with equal total weight with respect to affine transformations. Its Lipschitz constant is* 1*. This result holds for any dimension* d *and any norm on the underlying space* $\mathbb{R}^d$*.*

Proof. Let $A = \{(p_i, \alpha_i)_{i=1,\dots,n}\}$, $B = \{(q_j, \beta_j)_{j=1,\dots,m}\} \in \mathbb{W}^{d,G}$ be two weighted point sets with equal total weight $G \in \mathbb{R}_{>0}$ in dimension $d \in \mathbb{N}$. The equivariance of the center of mass under affine transformations is well-known. We prove the Lipschitz continuity. This proof already appeared in the work by Cohen and Guibas [21], and Rubner, Tomasi and Guibas [44] as a proof of a lower bound on the EMD. We have to show that

$$\|C(A) - C(B)\| \leq \mathrm{EMD}(A, B).$$

Let $F = \{f_{ij}\}_{i=1,\dots,n, j=1,\dots,m}$ be a flow determining $\mathrm{EMD}(A, B)$. Then

$$\begin{aligned}\|C(A) - C(B)\| &= \left\| \frac{1}{G}\sum_{i=1}^{n} \alpha_i p_i - \frac{1}{G}\sum_{j=1}^{m} \beta_j q_j \right\| \\ &= \frac{1}{G}\left\| \sum_{i=1}^{n} \alpha_i p_i - \sum_{j=1}^{m} \beta_j q_j \right\|.\end{aligned}$$

Using the flow conditions of Lemma 3.1 we get

$$\begin{aligned}\|C(A) - C(B)\| &= \frac{1}{G}\left\| \sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} p_i - \sum_{j=1}^{m}\sum_{i=1}^{n} f_{ij} q_j \right\| \\ &= \frac{1}{G}\left\| \sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} p_i - \sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} q_j \right\| \\ &= \frac{1}{G}\left\| \sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} (p_i - q_j) \right\| \\ &\leq \frac{1}{G}\sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} \|p_i - q_j\| \\ &= \mathrm{EMD}(A, B).\end{aligned}$$

The lower bound of 1 on the Lipschitz constant follows by Theorem 1.4 together with Theorem 3.1. □

3.3.3 A Uniqueness Result

We show that the center of mass as an EMD-reference point with respect to affine transformations is unique for weighted point sets with exactly 3 points and equal weight on each point. The proof follows a technique used by Knauer [39]. He proves that there is no $D_{\mathcal{H}}$-reference point with respect to affine transformations, where $D_{\mathcal{H}}$ denotes the Hausdorff distance. The strategy of his proof is the following: He first shows that, if there exists a $D_{\mathcal{H}}$-reference point for affine transformations, it has to map an arbitrary triangle to the center of mass of the vertices of its convex hull. Then he proves that the center of mass is not Lipschitz continuous on the set of triangles and therefore cannot be a $D_{\mathcal{H}}$-reference point for this set and every set containing the set of triangles.

Our proof uses the first part. We show that a reference point with respect to affine transformations of a weighted point set with 3 points and equal weight each has to be the center of mass. Since the center of mass is a Lipschitz continuous mapping on weighted point sets with respect to the EMD, we have proven that the center of mass is a unique reference point. The proof holds for weighted point sets in the plane. An extension to higher dimensions is straightforward.

Theorem 3.5. *Let $r\colon \mathbb{W}^{2,G} \to \mathbb{R}^2$ be an* EMD*-reference point with respect to affine transformations. Let $\alpha = G/3$ and $A = \{(p_1, \alpha), (p_2, \alpha), (p_3, \alpha)\} \in \mathbb{W}^{2,G}$ be a weighted point set with* 3 *points. Then, the reference point of A equals the center of mass of this set, i.e., $r(A) = C(A)$.*

Proof. Let $\Delta = \{(q_i, \alpha)\}_{i=1,\dots,3} \in \mathbb{W}^{2,G}$ be a weighted point set where the coordinates of the points are the vertices of an equilateral triangle in counterclockwise order. Let R be the counterclockwise rotation by $2\pi/3$ around the center of mass. Let Δ_R be the image of Δ under this rotation. The geometry and the weights of Δ_R and Δ are equal and we have that $\mathrm{EMD}(\Delta, \Delta_R) = 0$. Using Lipschitz continuity we see that $r(\Delta) = r(\Delta_R)$. Then

$$r(\Delta) \;=\; r(\Delta_R) \;=\; r(R(\Delta)) \;=\; R(r(\Delta)),$$

using the equivariance of r. Therefore, $r(\Delta)$ is a fixpoint under R and, since the center of mass $C(\Delta)$ is the only fixpoint of R, it follows that $r(\Delta) = C(\Delta)$. To prove the lemma, we consider the weighted point set A as the image of Δ under some affine transformation F and get

$$r(A) \;=\; r(F(\Delta)) \;=\; F(r(\Delta)) \;=\; F(C(\Delta)).$$

Since the center of mass is invariant under affine transformations, we have

$$r(A) \;=\; F(C(\Delta)) \;=\; C(F(\Delta)) \;=\; C(A).$$

□

We use the last theorem to prove the following result:

Theorem 3.6. *There are no* EMD*-reference points with respect to every transformation class containing the projective transformations.*

Proof. The class of projective transformations contains all affine transformations. Therefore, the only candidate for an EMD-reference point for weighted point sets with 3 weighted points and equal weight on each point is the center of mass according to Theorem 3.5. Since the center of mass is not equivariant under projective transformations, the theorem follows. □

3.3.4 Fermat-Weber Point

A candidate for a reference point on weighted point sets is the so-called Fermat-Weber point. This point minimizes the sum of the weighted distances to all weighted points.

Definition 3.4 (Fermat-Weber Point)**.** Let $A = \{(p_i, \alpha_i)\}_{i=1,\dots,n} \in \mathbb{W}^{d,G}$ be a weighted point set. Then

$$\mathrm{FW}(A) \;=\; \arg\min_{p\in\mathbb{R}^d} \sum_{i=1}^{n} \alpha_i \|p_i - p\|$$

is called the Fermat-Weber point of A.

Unfortunately, the Fermat-Weber point does not fulfill the Lipschitz continuity condition. As the proof shows, this even holds in the case of equal total weight.

Lemma 3.2. *The Fermat-Weber point is not an* EMD*-reference point for weighted point sets. This holds in any dimension and for any norm on the underlying space* $\mathbb{R}^d$*.*

Proof. Let O denote the origin and e_1 the first unit vector in $\mathbb{R}^d$. Consider the following two weighted point sets, where $0 < \delta < 1$:

$$\begin{aligned} A &= \{(O, 1-\delta), (3e_1, 1-\delta), (e_1, 2\delta)\} \\ B &= \{(O, 1-\delta), (3e_1, 1-\delta), (2e_1, 2\delta)\} \end{aligned}$$

Obviously, $\mathrm{FW}(A) = e_1$ and $\mathrm{FW}(B) = 2e_1$. Therefore $\|\mathrm{FW}(A) - \mathrm{FW}(B)\| = \|e_1\|$. Further, $\mathrm{EMD}(A, B) = \delta\|e_1\|$. Assuming that the Fermat-Weber point is Lipschitz continuous, there is a constant $c > 0$, such that

$$\|\mathrm{FW}(A) - \mathrm{FW}(B)\| \;\le\; c \cdot \mathrm{EMD}(A, B) \;\Leftrightarrow\; \|e_1\| \;\le\; c\delta\|e_1\|.$$

If now δ tends to zero, c has to be arbitrarily large. □

Remark. The example uses more than two points on a line. The same example holds if we disturb the inner points slightly to the top, thereby getting two weighted point sets with points in general position showing the non-Lipschitz continuity of the Fermat-Weber point.

In the last lemma we have shown that the Fermat-Weber point is not a reference point. Since being a reference point is only a sufficient condition to induce an approximation algorithm, it may still be possible that this point leads to a constant-factor approximation, namely that the Fermat-Weber point is a weak reference point. We show that this is not true:

Example. The position of A and B in the last proof easily shows that

$$\mathrm{EMD}^{\mathrm{opt}}(A, B) \;\le\; \delta\|e_1\|.$$

On the other hand, matching the Fermat-Weber points of the sets leads to

$$\mathrm{EMD}^{\mathrm{FW}}(A, B) \;=\; \|e_1\|(1-\delta).$$

Therefore

$$\frac{\mathrm{EMD}^{\mathrm{FW}}(A, B)}{\mathrm{EMD}^{\mathrm{opt}}(A, B)} \;\ge\; \frac{1-\delta}{\delta},$$

which tends to infinity as δ tends to 0.

The proofs above depend on placing a point with small weight at a suitable position. In the same way, or similarly by placing points with large weights, we can prove that the lower left corner and the center of mass of the boundary of the convex hull are not EMD-reference points or weak reference points. We generalize this result in the following section.

3.3.5 Non-Existence of Weak Reference Points Independent of Weights

Lemma 3.3. *There is no weak* EMD*-reference point independent of the weights of the points. This holds in any dimension $d \geq 2$ and for any norm on the underlying space $\mathbb{R}^d$.*

Proof. Assume there is a weak reference point $r\colon \mathbb{W}^{d,2} \to \mathbb{R}^d$ independent of the weights of the points. Let O denote the origin, and e_1, e_2 the first and second unit vector in $\mathbb{R}^d$, respectively. Define the two sets $A, B \in \mathbb{W}^{d,2}$ as

$$\begin{aligned} A &:= \{(O,1),(e_1,1)\}, \\ B &:= \{(O,1),(e_2,1)\}. \end{aligned}$$

Let $\tau := r(A) - r(B)$ and $B' := B + \tau$. Note that the weak reference points of A and B' coincide. There is at least one point $a \in A$ and one point $b' \in B'$ with $a \neq b'$. W.l.o.g. let those points be a_1 and $\tau(b_1)$. Since modifying the weights of the points does not affect the coordinates of $r(A)$ and $r(B)$, we can modify the weights of a_1 and b_1 to $2-\varepsilon$, where $0 < \varepsilon < 2$, without changing the positions of the weak reference points. We further change the weights of a_2 and b_2 to ε. With a slight abuse of notation we write A and B' for the modified sets. Then,

$$\mathrm{EMD}^{\mathrm{opt}}(A, B') \;\leq\; \frac{1}{2}\,\|e_1 - e_2\|\,\varepsilon,$$

since matching the two points with weight $2-\varepsilon$ leads to this value. On the other hand,

$$\mathrm{EMD}^{\mathrm{ref}}(A, B') \;\geq\; \frac{1}{2}\,\|a_1 - \tau(b_1)\|\,(2-2\varepsilon),$$

since at least the weight of $2-2\varepsilon$ has to be moved from a_1 to $\tau(b_1)$. Therefore,

$$\frac{\mathrm{EMD}^{\mathrm{ref}}(A, B')}{\mathrm{EMD}^{\mathrm{opt}}(A, B')} \;\geq\; \frac{\|a_1 - \tau(b_1)\|\,(2-2\varepsilon)}{\|e_1 - e_2\|\,\varepsilon},$$

which tends to ∞ as ε tends to 0. □

3.4 Approximation Using EMD-Reference Points

In this section we give approximation algorithms for the EMD under translations, rigid motions and similarities. The section is organized as follows: In each part we consider a class of transformations, construct an approximation algorithm for matching under these transformations for general EMD-reference points, and finally use the center of mass to obtain a concrete algorithm. These results hold only for weighted point sets with equal total weight.

In the following let $A = \{(p_i, \alpha_i)_{i=1,\ldots,n}\}$ and $B = \{(q_j, \beta_j)_{j=1,\ldots,m}\} \in \mathbb{W}^{d,G}$ be two weighted point sets in dimension d with equal total weight $G \in \mathbb{R}_{>0}$. Further let $r\colon \mathbb{W}^{d,G} \to \mathbb{R}^d$ be an EMD-reference point for weighted point sets with respect to the considered class of transformations and with Lipschitz constant c. Let $T^{\mathrm{ref}}(n)$ be the time to compute the reference point of a point set of size n, and $T^{\mathrm{EMD}}(n, m)$ the time to compute the EMD between two point sets of size n and m, respectively. Further, let $T^{\mathrm{rot}}(n, m)$ be the time to find a rotation R around a fixed point minimizing the EMD between two point sets of size n and m, respectively. Even for this restricted problem no exact and efficient algorithm is known so far.

An upper bound on $T^{\mathrm{EMD}}(n, m)$ is $O((nm)^2 \log(n+m))$ using a strongly polynomial minimum cost flow algorithm by Orlin [41]. See Section 3.2 for more details concerning the runtime $T^{\mathrm{EMD}}(n, m)$.

3.4.1 Translations

In general, the problem to compute the minimum EMD under translations exactly seems to be computationally expensive. In the following section we see that we can solve this problem in linear time if the two weighted point sets are weighted real numbers sorted by their coordinates. Otherwise we sort the numbers first which increases the runtime to $O(n \log n)$.

Translations on the Line

Theorem 3.7. *Let A, $B \in \mathbb{W}^{1,G}$ be two weighted point sets for some $G \in \mathbb{R}_{>0}$ sorted by their coordinates. Then their minimum* EMD *under translations can be computed in linear time.*

Proof. Let $A := \{(p_i, \alpha_i)_{i=1,\dots,n}\}$ and $B := \{(q_j, \beta_j)_{j=1,\dots,m}\}$ be two weighted point sets for some $G \in \mathbb{R}_{>0}$ sorted by their coordinates. Cohen and Guibas [20] prove that flow variables defining the EMD between A and B can be computed by the following greedy algorithm. Let $f_{ij} := 0$ for all i and j.

Algorithm 3.1.

1. $i := 1; \quad j := 1; \quad x := \alpha_1; \quad y := \beta_1.$
2. $f_{ij} := \min\{x, y\}.$
3. $x := x - f_{ij}; \quad y := y - f_{ij}.$
4. If ($x = 0$ and $i < n$): $i := i + 1$ and $x := \alpha_i$.
5. If ($x = 0$ and $i = n$): Stop.
6. If $y = 0$: $j := j + 1$ and $y := \beta_j$.
7. Goto 2.

In every loop of this algorithm at least one of the variables i and j is increased. Therefore, the number of loops performed is at most $n+m-1$ and the runtime of this algorithm is $O(n+m)$. Furthermore, in each loop exactly one flow variable gets a non-zero value. Thus, in the end there are at most $n + m - 1$ non-zero flow variables and they define the EMD between the sets A and B. We omit a formal proof of the correctness here. Cohen and Guibas [20] prove the result using cumulative distribution functions. The same result is also implied by the fact that the distances between the points fulfill the Monge property, and therefore the greedily chosen flow variables are optimal, see the work of Bein, Brucker, Park and Pathak [12].

Observing the above algorithm we see that the assignment to the flow variables only depends on the weights and the order of the points in their respective point set, and is independent of the position of the point sets on the real line. That is, this assignment to the flow variables is invariant under translations of the weighted point sets A and B.

Let $(p^{(k)}, q^{(k)})_{k=1,\dots,r}$ denote the at most $n+m-1$ pairs of points (a_i, b_j) with $f_{ij} > 0$, and let $g_k := f_{ij}$ denote the flow between those points. Let $f_k(\tau)$ denote the distance of $p^{(k)}$ and $q^{(k)}$ under the translation τ of B, i.e., $f_k(\tau) = |p^{(k)} - q^{(k)} - \tau|$. By the above remarks we know that $\text{EMD}(A, B+\tau) = \sum_{k=1}^{r} g_k f_k(\tau)$.

The functions f_k are convex. Moreover, they are differentiable if and only if $p^{(k)} \neq q^{(k)} + \tau$. Therefore, the function $\text{EMD}(A, B+\tau)$ as a weighted sum of these functions is convex, since all the weights g_k are positive. Further, $\text{EMD}(A, B+\tau)$ is differentiable if and only if $p^{(k)} \neq q^{(k)}+\tau$ for all $k = 1, \dots, r$. The derivative of f_k is -1 if $p^{(k)} < q^{(k)} + \tau$ and $+1$ if $p^{(k)} > q^{(k)} + \tau$. Therefore, the derivative of $\text{EMD}(A, B+\tau)$ is constant on intervals where $p^{(k)} \neq q^{(k)} + \tau$ for all $k = 1, \dots, r$. It follows that the minimum might be assumed on an interval, but there is at least one k where $p^{(k)} = q^{(k)} + \tau$ and the minimum is assumed. See the example below for an illustration of the functions f_k and the EMD under translations.

Unfortunately we cannot afford to compute the EMD at all r event points $\tau_k := p^{(k)} - q^{(k)}$. Instead we use the next recursion to compute the minimum EMD under translations.

We start with the list L of all r event points. We first determine the median τ^* of these points and its left and right neighbors $\tau_l := \sup_{\tau \in L}\{\tau < \tau^*\}$ and $\tau_r := \inf_{\tau \in L}\{\tau > \tau^*\}$. Assume that both exist. We can determine the median in linear time using a result by Blum, Floyd, Pratt, Rivest and Tarjan [14]. Let $\text{Lin}(x)$ denote a linear function and be initialized with $\text{Lin}(x) = 0$. We then compute $\text{EMD}(A, B+\tau^*) = \text{Lin} + \sum_{\tau \in L} g_\tau |\tau - \tau^*|$, where g_τ denotes the flow between the two points inducing this event point τ. We also compute $\text{EMD}(A, B+\tau_l)$ and $\text{EMD}(A, B+\tau_r)$ in the same way. This can be done in $O(|L|)$ time. By convexity we know that if both $\text{EMD}(A, B+\tau^*) \leq \text{EMD}(A, B+\tau_l)$ and $\text{EMD}(A, B+\tau^*) \leq \text{EMD}(A, B+\tau_r)$, then τ^* is optimal. Otherwise assume $\text{EMD}(A, B+\tau_l) < \text{EMD}(A, B+\tau^*)$. Again by convexity we know that the optimal τ has to lie on the left of τ^*. Thus we substitute L by $L' := \{\tau \in L \;:\; \tau < \tau^*\}$. Since τ^* is the median of L, $|L'| \leq |L|/2$.

We have to take care that in deeper recursion steps we can still compute the EMD in $O(|L|)$ time. We do this using the following observation: For any point $\tau' \in L'$ we have that $\text{EMD}(A, B+\tau') = \sum_{\tau \in L'} g_\tau |\tau - \tau'| + \sum_{\tau \in L \setminus L'} g_\tau (\tau - \tau')$. This is easy to see, since in deeper recursion steps the second sum is only computed at points in L'. These points are to the left of all points in $L \setminus L'$. Thus the distance can be represented by the left ray of the absolute value, which is a linear function.

Hence we add the second sum to $\text{Lin}(x)$ and proceed. In the following recursion step we can compute $\text{EMD}(A, B+\tau')$ as $\text{Lin} + \sum_{\tau \in L'} g_\tau |\tau - \tau'|$, which needs $O(|L'|)$ time.

The runtime of each recursion step is $O(|L|)$ and the length of the list in the following recursion step will be at most $|L|/2$. Thus, the runtime of the algorithm is described by the recursion $T(n) = O(n) + T(n/2)$ which resolves to $O(n)$.

We state the algorithm described above in pseudo-code.

Algorithm 3.2 (FindMinimum(L, Lin)).

1. If $|L| < 3$: Output $\min_{\tau \in L}\{\text{EMD}(A, B + \tau\}$. Stop.
2. Find median $\tau^* \in L$.
3. Find left neighbor $\tau_l := \sup_{\tau \in L}\{\tau < \tau^*\}$.
4. Find right neighbor $\tau_r := \inf_{\tau \in L}\{\tau > \tau^*\}$.
5. If $\tau_l = -\infty$ and $\tau_r = \infty$: Output $\text{EMD}(A, B + \tau^*)$. Stop.
6. If $\tau_l = -\infty$ and $\tau_r \neq \infty$:
   (a) If $\text{EMD}(A, B+\tau^*) \leq \text{EMD}(A, B+\tau^r)$: Output $\text{EMD}(A, B+\tau^*)$. Stop.
   (b) For all $\tau \in L$ with $\tau \leq \tau^*$: Lin := Lin $+g_\tau(x - \tau)$
   (c) $L' := \{\tau \in L \ : \ \tau > \tau^*\}$.
   (d) FindMinimum(L', Lin)
7. If $\tau_l \neq -\infty$ and $\tau_r = \infty$:
   (a) If $\text{EMD}(A, B+\tau^*) \leq \text{EMD}(A, B+\tau^l)$: Output $\text{EMD}(A, B+\tau^*)$. Stop.
   (b) For all $\tau \in L$ with $\tau \geq \tau^*$: Lin := Lin $+g_\tau(\tau - x)$
   (c) $L' := \{\tau \in L \ : \ \tau < \tau^*\}$.
   (d) FindMinimum(L', Lin)
8. If $\tau_l \neq -\infty$ and $\tau_r \neq \infty$:
   (a) If $\text{EMD}(A, B + \tau^*) \leq \text{EMD}(A, B + \tau^r)$ and $\text{EMD}(A, B + \tau^*) \leq \text{EMD}(A, B + \tau^l)$: Output $\text{EMD}(A, B + \tau^*)$. Stop.
   (b) If $\text{EMD}(A, B + \tau^l) < \text{EMD}(A, B + \tau^*)$:
      i. For all $\tau \in L$ with $\tau \geq \tau^*$: Lin := Lin $+g_\tau(\tau - x)$
      ii. $L' := \{\tau \in L \ : \ \tau < \tau^*\}$.
      iii. FindMinimum(L', Lin)
   (c) If $\text{EMD}(A, B + \tau^r) < \text{EMD}(A, B + \tau^*)$:
      i. For all $\tau \in L$ with $\tau \leq \tau^*$: Lin := Lin $+g_\tau(x - \tau)$
      ii. $L' := \{\tau \in L \ : \ \tau > \tau^*\}$.
      iii. FindMinimum(L', Lin)

□

Example. Let the weighted point sets A and B be given by $\{p_1, p_2, p_3\} = \{3, 4, 6\}$ and $\{q_1, q_2, q_3\} = \{5, 7, 12\}$. Then, the functions $f_i(\tau) = |p_i - q_i + \tau|$ for $i = 1, 2, 3$ describe the distances of p_i and q_i under the translation τ. Figure 3.1 illustrates f_1, f_2 and f_3 by the red, green and yellow graph, respectively. The blue graph describes the EMD under τ.

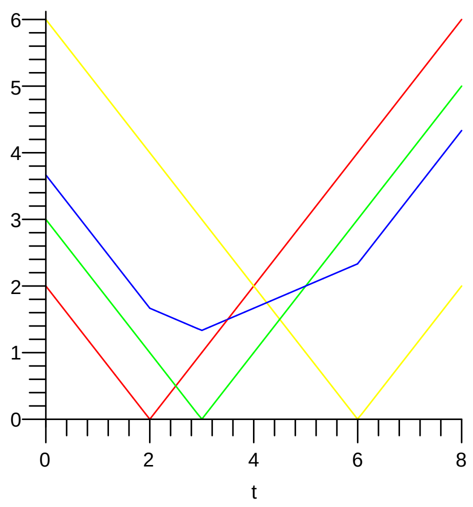


Figure 3.1: Illustration of the functions $f_i(\tau)$ and the EMD defined in the Example.

The General Case

In arbitrary dimension $d \geq 1$ we can apply our reference point framework to obtain an approximation on the minimum EMD under translations:

Algorithm 3.3.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$. Let B' be the image of B.
2. Output B' together with the distance $\text{EMD}(A, B')$.

The following theorem is a direct consequence of the result on the abstract approximation Algorithm 1.1 for translations, and Theorem 3.1.

Theorem 3.8. *Let $G \in \mathbb{R}_{>0}$ and let $r\colon \mathbb{W}^{d,G} \to \mathbb{R}^d$ be an* EMD*-reference point with respect to translations and with Lipschitz constant c. Algorithm 3.3 finds an approximately optimal matching for translations with approximation factor $c+1$ in time $O(T^{\text{ref}}(\max\{n, m\}) + T^{\text{EMD}}(n, m))$. This holds for arbitrary dimension d and any norm on the underlying space $\mathbb{R}^d$.*

Applying the center of mass as an EMD-reference point leads to an almost trivial approximation algorithm for the EMD under translations. Cohen and Guibas [21] propose the same method if the EMD is based on the squared Euclidean distance. In this case, the algorithm even finds the optimal translation.

Corollary 3.2. *Algorithm 3.3 using the center of mass as an* EMD*-reference point induces an approximation algorithm with approximation factor 2. Its runtime is $O(T^{\text{EMD}}(n, m))$.*

Proof. The Lipschitz constant of the center of mass as an EMD-reference point is 1 and this point can be computed in $O(\max\{n, m\})$ time. The overall runtime is dominated by the time to compute the Earth Mover's Distance. □

Lower Bound for Algorithm 3.3

We presented the center of mass as an EMD-reference point with Lipschitz constant 1, thus inducing an approximation algorithm for translations with approximation factor 2. We show that this bound cannot be improved in the Euclidean case. Recall that we write EMD_2 for the EMD based on the Euclidean distance.

Theorem 3.9. *There are weighted point sets where the upper bound on the approximation factor for Algorithm 3.3 using the center of mass as an* EMD_2*-reference point is assumed in the limit.*

Proof. Consider the following two weighted point sets in the plane, where $K \in \mathbb{N}$ is some integer.

$$\begin{aligned} A &:= \{((0,0),1),((1,0),K)\} \\ B &:= \{((0,0),1),((0,1),K)\} \end{aligned}$$

See Figure 3.2 for an illustration of A and B when the two centers of mass coincide.

We show that ${\text{EMD}_2}^C(A,B)/{\text{EMD}_2}^{\text{opt}}(A,B) \to 2$ as $K \to \infty$. Here, ${\text{EMD}_2}^C(A,B)$ denotes the EMD_2 between A and a translated version of B where the two centers of mass coincide, and ${\text{EMD}_2}^{\text{opt}}(A,B)$ denotes the EMD_2 of A and B under an optimal translation.

1. Calculation of ${\text{EMD}_2}^C(A,B)$: We first calculate the centers of mass of both sets. By definition

   $$C(A) = \frac{1}{W^A}\sum_{i=1}^{n}\alpha_i p_i = \frac{1}{K+1}\left((0,0)^T + K\,(1,0)^T\right) = \frac{K}{K+1}\,(1,0)^T.$$

   Similarly,

   $$C(B) = \frac{K}{K+1}\,(0,1)^T.$$

   See Figure 3.2 for an illustration of the matching according to the centers of mass.

   Using basic network flow theory, we know by the integrality of all weights that there is always an integral flow inducing the minimum cost flow. Thus it is an easy observation that there are two candidates for an assignment to the flow variables, see Figures 3.3 and 3.4. We compute the flow value in both cases.

   For the following calculations, note that the distance between a_1 and the center of mass $C(A)$ is $K/(K+1)$ and the distance between a_2 and $C(A)$ is $1 - K/(K+1) = 1/(K+1)$.

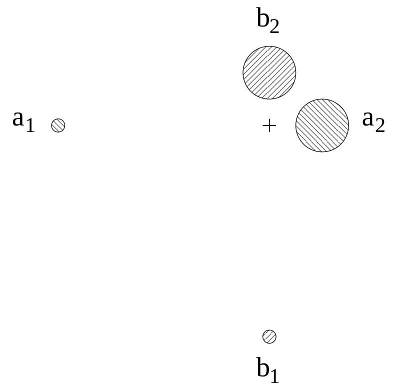


Figure 3.2: Matching of A, B according to their centers of mass.

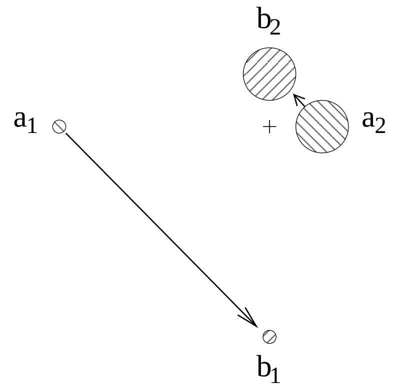


Figure 3.3: Flow in Version (a).

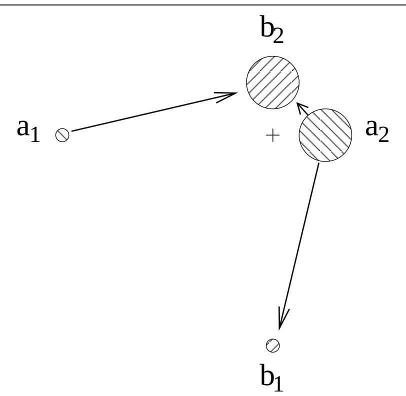


Figure 3.4: Flow in Version (b).

(a) $f_{11} = 1$, $f_{12} = 0$, $f_{21} = 0$ and $f_{22} = K$, see Figure 3.3:

$$\begin{aligned}
\mathrm{EMD}_2{}^C(A,B) &\leq \frac{1}{K+1}\left(K\sqrt{2\left(\frac{1}{K+1}\right)^2} + \sqrt{2\left(\frac{K}{K+1}\right)^2}\right) \\
&= \frac{1}{K+1}\left(\frac{K}{K+1}\sqrt{2} + \frac{K}{K+1}\sqrt{2}\right) \\
&= \frac{K}{(K+1)^2}\,2\sqrt{2} \\
&= \frac{2\sqrt{2}}{K}\left(1 + O\left(\frac{1}{K}\right)\right) \quad \text{as } K \text{ tends to } \infty
\end{aligned}$$

(b) $f_{11} = 0$, $f_{12} = 1$, $f_{21} = 1$ and $f_{22} = K - 1$, see Figure 3.4:

$$\begin{aligned}
\mathrm{EMD}_2{}^C(A,B) &\leq \frac{1}{K+1}\left(2\sqrt{\left(\frac{K}{K+1}\right)^2 + \left(\frac{1}{K+1}\right)^2} + (K-1)\sqrt{2\left(\frac{1}{K+1}\right)^2}\right) \\
&= \frac{1}{K+1}\left(2\sqrt{\frac{K^2+1}{(K+1)^2}} + (K-1)\frac{1}{K+1}\sqrt{2}\right) \\
&= \frac{1}{K+1}\left(2\sqrt{\frac{K-1}{K+1}} + \frac{K-1}{K+1}\sqrt{2}\right) \\
&= \frac{2+\sqrt{2}}{K}\left(1 + O\left(\frac{1}{K}\right)\right) \quad \text{as } K \text{ tends to } \infty
\end{aligned}$$

Therefore, for large K the flow variables given in (a) induce the minimum cost flow, i.e., $\mathrm{EMD}_2{}^C(A,B) = 2\sqrt{2}\,K/(K+1)^2$ for K large enough.

2. $\mathrm{EMD}_2{}^{\mathrm{opt}}(A,B)$: We do not calculate this distance exactly but find an upper bound by fixing the translation where a_2 and b_2 coincide, see Figure 3.5. It follows that

$$\mathrm{EMD}_2{}^{\mathrm{opt}}(A,B) \leq \frac{1}{K+1}\sqrt{2}.$$

Combining 1. and 2. we see that

$$\frac{\mathrm{EMD}_2{}^C(A,B)}{\mathrm{EMD}_2{}^{\mathrm{opt}}(A,B)} \geq \frac{\frac{K}{(K+1)^2}\,2\sqrt{2}}{\frac{1}{K+1}\sqrt{2}} = \frac{2K}{K+1} \rightarrow 2 \text{ as } K \rightarrow \infty.$$

□

Remark. The proof is independent of the considered diameter of the point set, so bounding the diameter does not lead to a better approximation. The proof depends on the weights of the points, exploiting an unbounded ratio of weights. The same proof works if we replace the points with weight K by K copies of unit weight points. Therefore, even with a bounded ratio of weights no improvement on the approximation factor is possible.

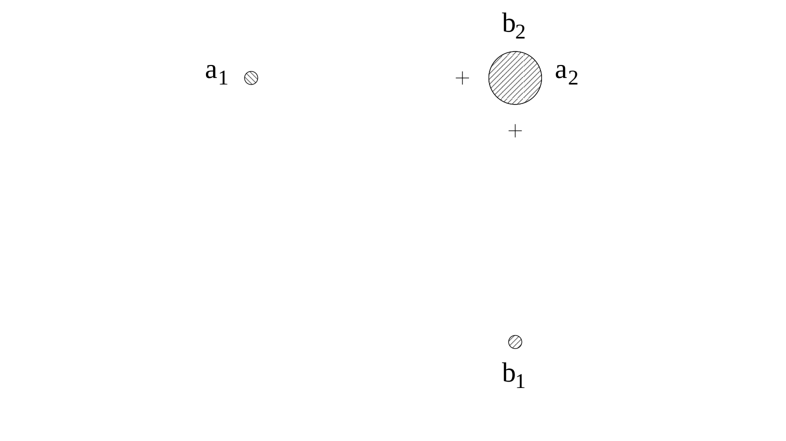


Figure 3.5: Translation where a_2 and b_2 coincide.

3.4.2 Fully Polynomial-Time Approximation Scheme for Translations

We can easily construct a $(1+\varepsilon)$-approximation for the EMD_p under translations, using the abstract algorithm from Section 1.5.2. This result, restricted to the Euclidean EMD in the plane, was already obtained by Cabello et al. [15]. The proof follows directly using Theorems 1.6 and 3.1 and is omitted.

Theorem 3.10. *Let $A, B \in \mathbb{W}^{d,G}$ for some $G \in \mathbb{R}_{>0}$ and dimension $d \in \mathbb{N}$ be two weighted point sets. Let $1 \leq p \leq \infty$. There exists an algorithm which for any $0 < \varepsilon < 1$ finds a translation τ^ε, such that*

$$\mathrm{EMD}_p(A, \tau^\varepsilon(B)) \;\leq\; (1+\varepsilon) \cdot \mathrm{EMD}_p(A, \tau^{\mathrm{opt}}(B)).$$

Its runtime is $O(\varepsilon^{-d} \cdot T^{\mathrm{EMD}_p}(n, m))$.

3.4.3 Rigid Motions

In the following we investigate the problem to compute the minimum EMD under rotations, rigid motions and later similarities. Thus we always assume that the dimension is at least 2. The following algorithm gives a first approach to obtain an approximation of the EMD under rigid motions, i.e., combinations of translations and rotations. This algorithm is not yet practical, since we have to find the optimal matching of two weighted point sets under rotations of one of those around a fixed point. We approximate this problem of finding the optimal rotation later and thereby create an efficient and implementable algorithm.

Algorithm 3.4.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.
2. Find an optimal matching of A and B' under rotations of B' around $r(A)$.
   Let B'' be the image of B' under this rotation.
3. Output B'' together with the distance $\mathrm{EMD}(A, B'')$.

Algorithm 3.4 equals the abstract Algorithm 1.2 if we use the optimal rotation. Thus the following theorem is a direct consequence of Theorems 1.7 and 3.1.

Theorem 3.11. *Let $G \in \mathbb{R}_{>0}$ and let $r\colon \mathbb{W}^{d,G} \to \mathbb{R}^d$ be an* EMD*-reference point with respect to rigid motions and with Lipschitz constant c. Algorithm 3.4 finds an approximately optimal matching for rigid motions with approximation factor $c+1$ in time $O(T^{\text{ref}}(\max\{n,m\}) + T^{\text{EMD}}(n,m) + T^{\text{rot}}(n,m))$. This holds for arbitrary dimension $d \geq 2$ and any norm on the underlying space $\mathbb{R}^d$.*

We apply the center of mass as an EMD-reference point to the last result:

Corollary 3.3. *Algorithm 3.4 using the center of mass as an* EMD*-reference point induces an approximation algorithm with approximation factor* 2 *and runtime $O(T^{\text{rot}}(n,m)+T^{\text{EMD}}(n,m))$. This holds for arbitrary dimension $d \geq 2$ and any norm on the underlying space $\mathbb{R}^d$.*

Proof. The Lipschitz constant of the center of mass as an EMD-reference point is 1. This point can be computed in linear time, which is dominated by the time to compute the EMD. □

Lower Bound for Algorithm 3.4

We show that the approximation factor given in the last corollary is tight in the Euclidean case, thus we prove the following theorem:

Theorem 3.12. *There are weighted point sets where the upper bound on the approximation factor for Algorithm 3.4 using the center of mass as an* EMD_2*-reference point is assumed in the limit.*

Proof. We prove the theorem by using similar weighted point sets as we use in the proof of the lower bound for Algorithm 3.3, see Figure 3.2 in Section 3.4.1. We introduce two more points for every set to prevent them from being rotated. These points are far away from the center of mass and have a low weight. One of them is above and one is below the original point set. They further have the same distance to the center of mass. Thereby the position of the center of mass is not affected. These points coincide when the sets given in the lower bound for Algorithm 3.3 are unrotated and have coinciding centers of mass. Because of the low weight, a short translation does not have a major effect on the resulting EMD. A small rotation instead immediately leads to a large distance between those points and therefore these two points contribute significantly. Let $\delta > 0$ be a fixed constant and let $\alpha_0 := \arcsin(\delta - \delta^2/4)$. The introduced points depend on a large parameter K and a small parameter $\varepsilon := 4/(K\sqrt{1-\cos\alpha_0})$. Note that ε tends to 0 as K tends to ∞.

Here are two point sets realizing the lower bound, see Figure 3.6 for an illustration.

$$\begin{aligned} A &:= \left\{((0,0),1),((1,0),K),\left(\left(\frac{K}{K+1},K\right),\varepsilon\right),\left(\left(\frac{K}{K+1},-K\right),\varepsilon\right)\right\} \\ B &:= \left\{\left(\left(\frac{K}{K+1},-\frac{K}{K+1}\right),1\right),\left(\left(\frac{K}{K+1},\frac{1}{K+1}\right),K\right),\right. \\ &\qquad \left.\left(\left(\frac{K}{K+1},K\right),\varepsilon\right),\left(\left(\frac{K}{K+1},-K\right),\varepsilon\right)\right\} \end{aligned}$$

We make a case distinction for clockwise rotations with angles $0 \leq \alpha \leq \alpha_0$ and $\alpha_0 \leq \alpha \leq \pi/2$. The remaining cases are similar and omitted here. Let R_α denote the clockwise rotation by α.

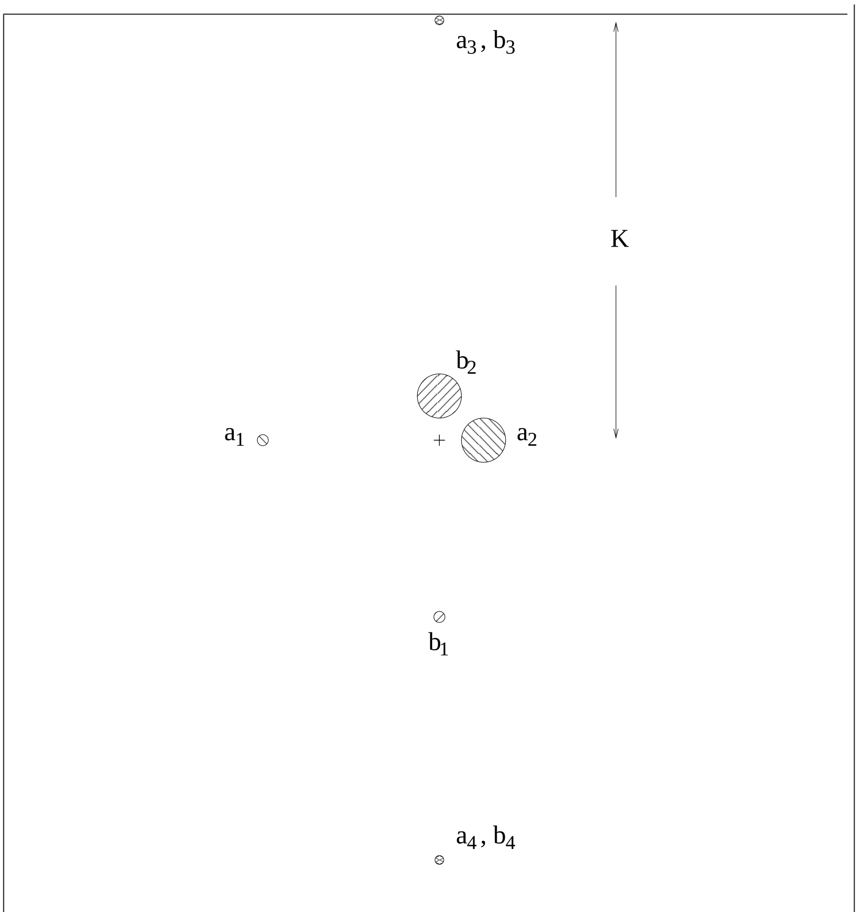


Figure 3.6: Matching of A and B according to their centers of mass.

We show that for any $\delta > 0$ we have that $\mathrm{EMD_2}^C(A,B)/\mathrm{EMD_2}^{\mathrm{opt}}(A,B) \geq 2-\delta$ as K tends to ∞. Here, $\mathrm{EMD_2}^{\mathrm{opt}}$ denotes the optimal EMD_2 under rigid motions and $\mathrm{EMD_2}^C$ the optimum where the centers of mass coincide. Using the same arguments as in the proof for the lower bound for Algorithm 3.3 in Section 3.4.1, we see that

$$\mathrm{EMD_2}^{\mathrm{opt}}(A,B) \;\leq\; \frac{1}{K+1+2\varepsilon}\left(\sqrt{2}+\frac{2\sqrt{2}\,\varepsilon}{K+1}\right)$$

by computing the EMD_2 of A and a translated version of B where the two points with weight K coincide.

Let $0 \leq \alpha \leq \alpha_0$. Since K is large and the rotation angle is small, a_3 will be matched to $R_\alpha(b_3)$ and a_4 to $R_\alpha(b_4)$. By similar considerations as before in the case of translations, we see that a_1 is matched to $R_\alpha(b_1)$ and a_2 to $R_\alpha(b_2)$. Easy calculation shows

$$\begin{aligned} \|a_1 - R_\alpha(b_1)\| &= \sqrt{2}\sqrt{1-\sin\alpha}\,K/(K+1) \\ \text{and} \quad \|a_2 - R_\alpha(b_2)\| &= \sqrt{2}\sqrt{1-\sin\alpha}\,/(K+1). \end{aligned}$$

Therefore

$$\mathrm{EMD_2}^C(A,R_\alpha(B)) \;\geq\; \frac{2\sqrt{2}\,K}{(K+1+2\varepsilon)(K+1)}\sqrt{1-\sin\alpha},$$

and

$$\frac{\mathrm{EMD_2}^C(A,R_\alpha(B))}{\mathrm{EMD_2}^{\mathrm{opt}}(A,B)} \quad\geq\quad \frac{\frac{2\sqrt{2}\,K}{K+1}\sqrt{1-\sin\alpha}}{\sqrt{2}+\frac{2\sqrt{2}\,\varepsilon}{K+1}},$$

which tends to $2\sqrt{1-\sin\alpha}$ as K tends to ∞. For any $\alpha \leq \alpha_0$ we have

$$2\sqrt{1-\sin\alpha} \;\geq\; 2\sqrt{1-\sin\alpha_0} \;=\; \sqrt{1-\delta+\delta^2/4} \;=\; 2(1-\delta/2) \;=\; 2-\delta.$$

Let $\alpha_0 \leq \alpha \leq \pi/2$. We make a case distinction on the flow variable f_{33}:

1. $f_{33} \geq \varepsilon/2$.
   It is easy to calculate that
   $$\|a_3 - R_\alpha(b_3)\| = \;\sqrt{2}\,K\sqrt{1-\cos\alpha}$$
   and therefore
   $$\mathrm{EMD_2}^C(A,R_\alpha(B)) \;\geq\; \frac{\sqrt{2}K\varepsilon/2\sqrt{1-\cos\alpha}}{K+1+2\varepsilon}.$$
   Thus
   $$\frac{\mathrm{EMD_2}^C(A,R_\alpha(B))}{\mathrm{EMD_2}^{\mathrm{opt}}(A,B)} \;\geq\; \frac{\sqrt{2}K\varepsilon/2\sqrt{1-\cos\alpha}}{\sqrt{2}+\frac{2\sqrt{2}\,\varepsilon}{K+1}},$$
   which tends to $K\varepsilon/2\sqrt{1-\cos\alpha} \geq K\varepsilon/2\sqrt{1-\cos\alpha_0} = 2$ as K tends to ∞.

2. $f_{33} < \varepsilon/2$.
   This immediately implies that $f_{31}+f_{32}+f_{34} \geq \varepsilon/2$ and $f_{13}+f_{23}+f_{34} \geq \varepsilon/2$. Therefore
   $$\mathrm{EMD_2}^C(A,R_\alpha(B)) \;\geq\; \frac{2(K-1)\varepsilon/2}{K+1+2\varepsilon}$$

as K tends to ∞. Thus

$$\frac{\mathrm{EMD_2}^C(A, R_\alpha(B))}{\mathrm{EMD_2}^{\mathrm{opt}}(A,B)} \geq \frac{2(K-1)\varepsilon/2}{\sqrt{2}+\frac{2\sqrt{2}\,\varepsilon}{K+1}},$$

which tends to $K\varepsilon/\sqrt{2} = 4/(\sqrt{2}\sqrt{1-\cos\alpha_0}\,) \geq 4/\sqrt{2} > 2$ as K tends to ∞.

□

3.4.4 Rigid Motion Approximation Using Rotation Approximation

Fixing the position of the coinciding EMD-reference points as the rotation center in Algorithm 3.4 eliminates several degrees of freedom and the problem to find the optimal rotation is easier than the one finding the optimal rigid motion itself. Unfortunately, even for this problem no efficient algorithm is known so far. Therefore we investigate approximation algorithms. The next lemma gives an approximation if the underlying norm is the Euclidean distance. This result was already obtained by Cabello et al. [15] for the Euclidean EMD in the plane. We extend the result to arbitrary dimension. After that we generalize the statement to all L_p-distances, where $1 \leq p \leq \infty$.

Lemma 3.4. *Let A, B be two weighted point sets with equal total weight $G \in \mathbb{R}_{>0}$ in dimension $d \geq 2$. Let $p^* \in \mathbb{R}^d$ be any point and let $\mathrm{Rot}(p^*)$ be the set of rotations around p^*. There exists a rotation $R' \in \mathrm{Rot}(p^*)$ such that*

$$\mathrm{EMD}_2(A, R'(B)) \leq 2 \cdot \min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_2\left(A, R(B)\right),$$

where R' aligns p^*, a point of A, and a point of B.

Proof. W.l.o.g. let $A = \{(p_i, \alpha_i)_{i=1,\dots,n}\}$ and $B = \{(q_j, \beta_j)_{j=1,\dots,m}\}$ be in optimal position with respect to rotations of B around p^*. Let $F^* := \{f^*_{ij}\}$ be a flow defining $\mathrm{EMD}_2(A,B)$. For all pairs of points $p_i \in A$ and $q_j \in B$, $p_i, p_j \neq p^*$, let R_{ij} be the smallest rotation around p^* aligning those two points. More precisely, let R_{ij} be the rotation with the following properties:

1. R_{ij} rotates around the $(d-2)$-dimensional subspace S_{ij} that contains p^* and is orthogonal to the plane E_{ij} spanned by the points p^*, p_i and q_j.
2. $R_{ij}(B)$ aligns p^*, p_i and q_j.
3. For the rotation angle $\phi(R_{ij})$ we have $|\phi(R_{ij})| \leq \pi$.

Note that the rotation angle is given by the smaller angle between the lines through p_i and p^*, and q_j and p^*. Thus this angle is defined in arbitrary dimension.

Let $\mathcal{R}$ be the set of all rotations R_{ij}. Let $R' \in \mathcal{R}$, such that for all $R \in \mathcal{R}$ we have $|\phi(R')| \leq |\phi(R)|$. If $\phi(R') = 0$, R' fulfills the claim of the lemma.

In the 2-dimensional case, Giannopoulos [26] shows that

$$\|R'(q_j) - p_i\|_2 \leq 2\,\|q_j - p_i\|_2 \tag{3.1}$$

for all points $p_i \in A$ and $q_j \in B$.

We prove that inequality (3.1) carries over to arbitrary dimension $d \geq 2$, see also Figure 3.7.

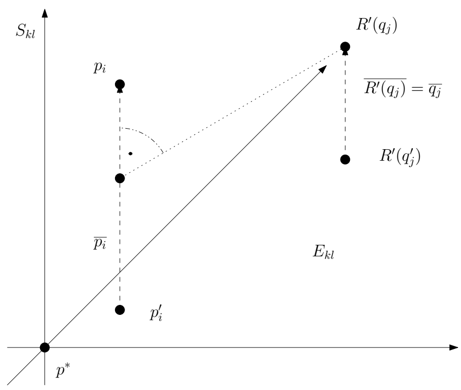


Figure 3.7: Construction in higher dimensions.

Let p_k and q_l be a pair of points inducing R' and let p_i and q_j be two arbitrary points. Let p'_i and p'_j be the orthogonal projections of p_i and q_j onto E_{kl}, respectively. Let $\overline{p_i} := p_i - p'_i$ and $\overline{q_j} := q_j - q'_j$. Now, $p_i = p'_i + \overline{p_i}$ and $q_j = q'_j + \overline{q_j}$. Since $\overline{q_j}$ is parallel to S_{kl} this vector is invariant under R' and thus we have $R'(q_j) = R'(q'_j + \overline{q_j}) = R'(q'_j) + \overline{q_j}$. Then,

$$\begin{aligned}
&\|R'(q_j) - p_i\|_2 \\
&= \|R'(q'_j) + \overline{q_j} - p'_i - \overline{p_i}\|_2 \\
&= \sqrt{\|R'(q'_j) - p'_i\|_2^2 + \|\overline{q_j} - \overline{p_i}\|_2^2}, \quad \text{by Pythagoras' Theorem} \\
&\leq \sqrt{4\,\|q'_j - p'_i\|_2^2 + \|\overline{q_j} - \overline{p_i}\|_2^2}, \quad \text{using the 2-dimensional case inside } E_{kl} \\
&\leq \sqrt{4\,\|q'_j - p'_i\|_2^2 + 4\,\|\overline{q_j} - \overline{p_i}\|_2^2} \\
&= 2\,\|q'_j + \overline{q_j} - p'_i - \overline{p_i}\|_2, \quad \text{by Pythagoras' Theorem} \\
&= 2\,\|q_j - p_i\|_2.
\end{aligned}$$

Following Cabello et al. [15] we prove the lemma in arbitrary dimension $d \geq 2$:

$$\begin{aligned}
\mathrm{EMD}_2(A, R'(B)) &= \frac{1}{G} \cdot \min_{F \in \mathcal{F}} \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}\,\|p_i - R'(q_j)\|_2 \\
&\leq \frac{1}{G} \sum_{i=1}^{n} \sum_{j=1}^{m} f^*_{ij}\,\|p_i - R'(q_j)\|_2 \\
&\leq \frac{1}{G} \sum_{i=1}^{n} \sum_{j=1}^{m} 2\,f^*_{ij}\,\|p_i - q_j\|_2 \\
&= 2 \cdot \mathrm{EMD}_2(A, B).
\end{aligned}$$

□

We generalize the last lemma to all L_p-distances with $1 \leq p \leq \infty$. Note that the approximation in the following lemma is induced by the same rotation R' for any L_p-distance.

Lemma 3.5. *Let $A, B \in \mathbb{W}^{d,G}$ for some $G \in \mathbb{R}_{>0}$ be two weighted point sets and $p^* \in \mathbb{R}^d$ be any fixed point. Let $1 \leq p \leq \infty$. Let $\mathrm{Rot}(p^*)$ be the set of rotations around p^*. There exists a rotation $R' \in \mathrm{Rot}(p^*)$ such that*

$$\mathrm{EMD}_p(A, R'(B)) \;\leq\; 2\sqrt{d} \cdot \min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_p(A, R(B)),$$

where R' aligns p^*, a point of A, and a point of B.

Proof. Let $1 \leq p < 2$. It is well-known that $\|x\|_2 \leq \|x\|_p \leq \sqrt{d}\,\|x\|_2$ for all $x \in \mathbb{R}^d$. Let R' be the rotation which gives the 2-approximation in the Euclidean case, see the proof of Lemma 3.4. Then,

$$\begin{aligned}
&\mathrm{EMD}_p(A, R'(B)) \\
=\;& \frac{1}{G} \cdot \min_{F=(f_{ij})} \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}\, \|a_i - R'(b_j)\|_p \\
\leq\;& \frac{1}{G} \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}^{(2)}\, \|a_i - R'(b_j)\|_p, \quad \text{where } f_{ij}^{(2)} \text{ induces } \mathrm{EMD}_2(A, R'(B)) \\
\leq\;& \frac{1}{G} \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}^{(2)} \sqrt{d}\, \|a_i - R'(b_j)\|_2, \quad \text{since } \|x\|_p \leq \sqrt{d}\,\|x\|_2 \\
\leq\;& 2\sqrt{d} \cdot \min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_2(A, R(B)), \quad \text{Lemma 3.4} \\
\leq\;& 2\sqrt{d} \cdot \mathrm{EMD}_2(A, R^{(p)}(B)), \quad \text{where } R^{(p)} := \arg\min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_p(A, R(B)) \\
=\;& \frac{2\sqrt{d}}{G} \cdot \min_{F=(f_{ij})} \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}\, \|a_i - R^{(p)}(b_j)\|_2 \\
\leq\;& \frac{2\sqrt{d}}{G} \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}^{(p)}\, \|a_i - R^{(p)}(b_j)\|_2, \quad \text{where } f_{ij}^{(p)} \text{ induces } \mathrm{EMD}_p(A, R^{(p)}(B)) \\
\leq\;& 2\sqrt{d} \cdot \min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_p(A, R(B)), \quad \text{since } \|x\|_2 \leq \|x\|_p.
\end{aligned}$$

This completes the proof for $1 \leq p < 2$. An analogous calculation using the fact that $\|x\|_p \leq \|x\|_2 \leq \sqrt{d}\,\|x\|_p$ for any $2 < p \leq \infty$ shows the other case. □

As we will see in Section 3.4.4, Lemmas 3.4 and 3.5 directly lead to an approximation algorithm for the EMD under rotations around a fixed point in the plane. We prove this by observing that in the plane fixing one point-to-point correspondence uniquely determines a rotation. In dimensions ≥ 3 this is not the case. There, $d-1$ correspondences are necessary. We use the following lemma in Section 3.4.4 to show that we can successively fix these correspondences and thereby find an approximation in higher dimensions.

Lemma 3.6. *Let A, B be two weighted point sets with equal total weight in dimension $d \geq 2$. Let S^* be a fixed d'-dimensional affine space with $0 \leq d' < d-1$. Let $A \setminus S^* \neq \emptyset$ and $B \setminus S^* \neq \emptyset$. Let $\mathrm{Rot}(S^*)$ be the set of rotations leaving S^* invariant. There exists a rotation $R' \in \mathrm{Rot}(S^*)$ such that*

$$\mathrm{EMD}_2(A, R'(B)) \;\leq\; 2 \cdot \min_{R \in \mathrm{Rot}(S^*)} \mathrm{EMD}_2(A, R(B)),$$

where R' rotates B such that S^*, at least one point of $A \setminus S^*$ and at least one rotated point of $B \setminus S^*$ are in a $(d'+1)$-dimensional affine space.

Proof. W.l.o.g. let A and B be in optimal position with respect to rotations around S^*. We prove the result following the technique of the proof of Lemma 3.4. Let $p^* \in S^*$ be a fixed point. Let $S^\perp := p^* + (S^*)^\perp$ be the orthogonal affine space containing p^*. For all points $p_i \in A \setminus S^*$ and $q_j \in B \setminus S^*$ let $p_i^\perp$ and $q_j^\perp$ be their orthogonal projections onto $S^\perp$. Let R_{ij} be the rotation with the following properties:

1. R_{ij} rotates around the $(d-2)$-dimensional subspace that contains S^* and is orthogonal to the plane E_{ij} spanned by the points p^*, $p_i^\perp$ and $q_j^\perp$.
2. $R_{ij}(B)$ aligns p^*, $p_i^\perp$ and $q_j^\perp$.
3. For the rotation angle $\phi(R_{ij})$ we have $-\pi < \phi(R_{ij}) \leq \pi$.

Note that the rotation R_{ij} is independent of the concrete choice of $p^* \in S^*$ since choosing a different point only causes a parallel translation of $p_i^\perp$ and $q_j^\perp$. Further, the rotation angle is given by the smaller angle between the lines through $p_i^\perp$ and p^*, and through $q_j^\perp$ and p^*. Thus this angle is defined in arbitrary dimension.

We conclude like in the proof of Lemma 3.4: We first determine a rotation R' around S^* which minimizes the absolute value of the rotation angle among all rotations R_{ij}. We further determine a pair of points p_k and q_l inducing R' and again bound the distance of any pair of points by using the orthogonal projections onto E_{kl} and applying Pythagoras' Theorem. Finally we bound the EMD_2 after rotation and the lemma follows. □

Rotation Approximation - A Better Bound

In the last section we have recapitulated and generalized a result by Cabello et al. [15] which states that there is a rotation aligning the rotation center, a point of A, and a point of B, such that the EMD_2 in this position is at most twice the optimum. In this section we indicate that we can improve this bound significantly using the same rotation and a different analysis.

First we show an upper bound on the approximation ratio of any pair of weighted point sets. This bound is a fraction of weighted sums of the distance of two pairs of points after and before rotation. We further restrict the possible positions of these points. We discretize this function in one variable and compute it. The result is a function with a maximum at approximately 1.155. This implies that the approximation ratio of 2 can be replaced by this value. Further, the improved approximation bound of 1.155 would carry over to higher dimensions and to similarities, in the same way as the bound of 2 does.

A final step missing in the proof may be bounding the partial derivative with respect to the discretized variable by a constant. By choosing the discretization fine enough we could bound the maximum achieved in the interval between to discretization points. Unfortunately, this step is missing so far. Anyway, the results in this section finally may lead to a formal proof of the new upper bound. We indicate the new upper bound for weighted point sets in the plane and consider the Euclidean case.

Let $A = \{a_1, \ldots, a_n\} = \{(p_i, \alpha_i)_{i=1,\ldots,n}\}$ and $B = \{b_1, \ldots, b_m\} = \{(q_j, \beta_j)_{j=1,\ldots,m}\} \in \mathbb{W}^{2,G}$ be weighted point sets in optimal position with respect to rotations around some point $p^* \in \mathbb{R}^d$. W.l.o.g. let p^* be the origin. Let $F^* = (f_{ij}^*)$ be a set of flow variables inducing the EMD_2 between A and B.

We want to find an upper bound on the approximation ratio of our approximation algorithm for rotations. Thus we want to find an upper bound on

$$\frac{\mathrm{EMD}_2(A, R_\phi(B))}{\mathrm{EMD}_2(A, B)},$$

where ϕ is the minimum angle between two points in A and B, see the construction used in the proof of Lemma 3.4. We bound this fraction from above by choosing the same flow variables in the rotated position:

$$\frac{\mathrm{EMD}_2(A, R_\phi(B))}{\mathrm{EMD}_2(A, B)} \;\le\; \frac{\sum_{i=1}^n \sum_{j=1}^m f_{ij}^* \, \|a_i - R_\phi(b_j)\|_2}{\sum_{i=1}^n \sum_{j=1}^m f_{ij}^* \, \|a_i - b_j\|_2}. \tag{3.2}$$

By substituting each weighted point $a_i \in A$ by a set of weighted points $\{(p_i, f_{ij}^*) \,:\, f_{ij}^* > 0\}$ and defining the new flow variables appropriately, we can assume that for each point $a_i \in A$ there is exactly one point $b_j \in B$ with $f_{ij}^* > 0$. By substituting the points in B analogously, we can assume that there is a one-to-one correspondence between the weighted points in A and B. Thus we can assume that $A = \{a_1, \ldots, a_k\} = \{(p_i, \alpha_i)_{i=1,\ldots,k}\}$ and $B = \{b_1, \ldots, b_k\} = \{(q_i, \alpha_i)_{i=1,\ldots,k}\}$. Additionally we have $f_{ij}^* = \alpha_i$ if $i = j$ and 0 otherwise. Note that the quotient (3.2) does not change and we have

$$\frac{\mathrm{EMD}_2(A, R_\phi(B))}{\mathrm{EMD}_2(A, B)} \;\le\; \frac{\sum_{i=1}^n \sum_{j=1}^m f_{ij}^* \, \|a_i - R_\phi(b_j)\|_2}{\sum_{i=1}^n \sum_{j=1}^m f_{ij}^* \, \|a_i - b_j\|_2} \;=\; \frac{\sum_{i=1}^k \alpha_i \, \|a_i - R_\phi(b_i)\|_2}{\sum_{i=1}^k \alpha_i \, \|a_i - b_i\|_2}.$$

For further simplicity and to keep notation easy, we modify the two weighted point sets in the following way: For every $1 \le i \le k$ we rotate the point b_i around the origin onto the y-axis. We rotate the corresponding point a_i by the same angle. Let $A' = \{a'_1, \ldots, a'_k\}$ and $B' = \{b'_1, \ldots, b'_k\}$ be the transformed sets. It is easy to observe that

$$\frac{\mathrm{EMD}_2(A, R_\phi(B))}{\mathrm{EMD}_2(A, B)} \;\le\; \frac{\sum_{i=1}^k \alpha_i \, \|a_i - R_\phi(b_i)\|_2}{\sum_{i=1}^k \alpha_i \, \|a_i - b_i\|_2} \;=\; \frac{\sum_{i=1}^k \alpha_i \, \|a'_i - R_\phi(b'_i)\|_2}{\sum_{i=1}^k \alpha_i \, \|a'_i - b'_i\|_2} \tag{3.3}$$

since we did not change the distances between the points a_i and b_i.

With a slight abuse of notation we omit the prime and denote the transformed sets by A and B. The two weighted point sets are in optimal position, since the original point sets A and B are in optimal position with respect to rotations and the modifications above did not change the distance between original points with non-zero flow. Rotating B while keeping A and α_i fixed will not decrease the cost. This optimality condition, which is formally derived below, has the following physical interpretation: The two point sets together with the one-to-one correspondence describe a system of forces. We can imagine that the points in A are fixed in the plane and pull at the points of B, where the points of B are fixed on a line through the origin and this line is free to rotate around the origin. The force between any pair of points equals the flow. This rotational system is in equilibrium. Therefore we have

$$\sum_{\{i \,:\, (a_i)_x > 0\}} \cos\gamma_i \, \alpha_i \, \|b_i\|_2 \;-\; \sum_{\{i \,:\, (a_i)_x < 0\}} \cos\gamma_i \, \alpha_i \, \|b_i\|_2 \;=\; 0, \tag{3.4}$$

where γ_i equals the angle between the x-axis and the line through a_i and b_i, see Figure 3.8.

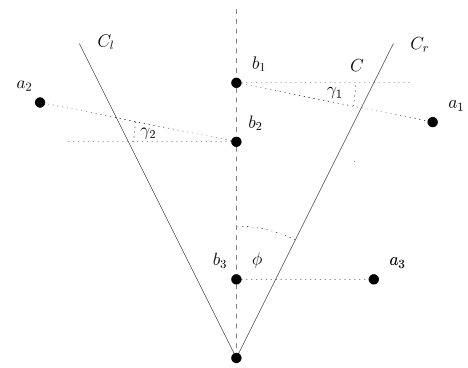


Figure 3.8: A and B after rotating the points of B to the y-axis.

We formally derive this equation: Let $f(\rho) := \sum_{i=1}^{k} \alpha_i \, \|a_i - R_\rho(b_i)\|_2$, where as usual the parameter ρ describes the rotation angle of the rotation R. Since A and B are in optimal position with respect to rotations, we have

$$f(0) \;=\; \min_{\rho} \sum_{i=1}^{k} \alpha_i \, \|a_i - R_\rho(b_i)\|_2$$

and thus $f'(0) = 0$. Let $b_i = (0, y_i)$. Then, $R_\rho(b_i) = (-y_i \sin\rho, y_i \cos\rho)$. By easy computation we see

$$f'(\rho) \;=\; \sum_{i=1}^{k} \alpha_i y_i \frac{(a_i)_x \cos\rho + (a_i)_y \sin\rho}{\|a_i - R_\rho(b_i)\|_2}$$

and therefore

$$f'(0) = 0 \quad \Leftrightarrow \quad \sum_{i=1}^{k} \alpha_i y_i \frac{(a_i)_x}{\|a_i - b_i\|_2} = 0$$

Since $\cos\gamma_i = |(a_i)_x| / \|a_i - b_i\|_2$, equation (3.4) follows.

Now, let C be the vertical cone with opening angle 2ϕ, see Figure 3.8. Let C_r and C_l denote the right and left ray of C, respectively. Note that there is no point of A inside the cone, since ϕ is the minimum angle aligning the origin, a point of A, and a point of B.

We first show that the upper bound still holds if the points of A with an x-coordinate greater than 0 lie on C_r, and the other points lie on C_l, see Figures 3.9 and 3.10. In this construction we leave the weights of the points unchanged.

Let $R, L \subset A$ denote the weighted points of A on the right and on the left of the y-axis, respectively. For given points $l \in L$ and $r \in R$ let b_l, b_r be the corresponding points in B. We further use α_l and α_r to denote their weights. Then, with r' denoting the point r shifted onto C_r along the segment $r\,b_r$, and with l' denoting the point l shifted onto C_l along $l\,b_l$, we

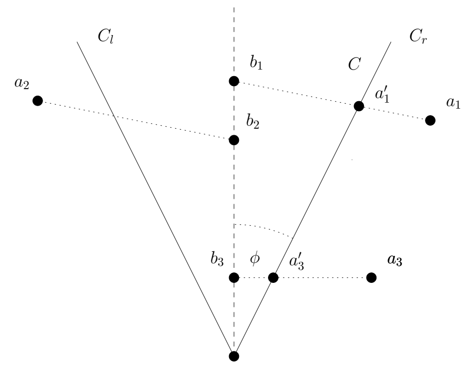


Figure 3.9: Shifting the points on the right onto C_r.

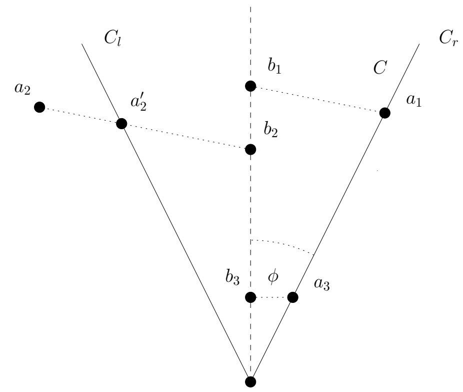


Figure 3.10: Shifting the points on the left onto C_l.

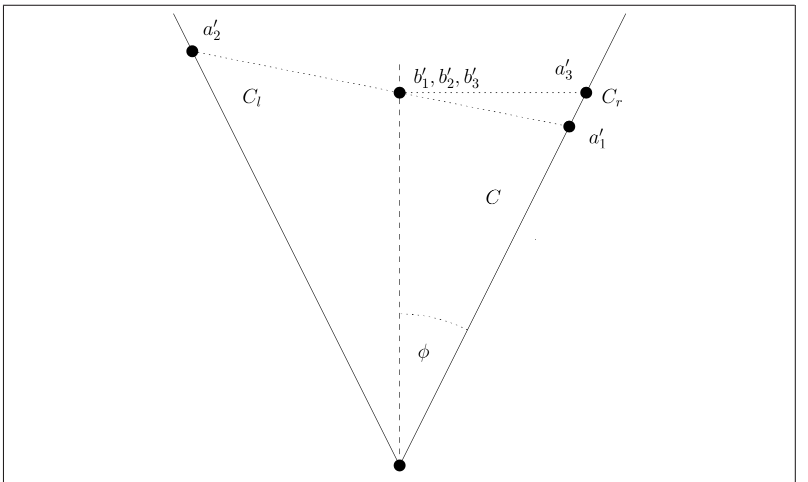


Figure 3.11: Using the Theorem of Intersecting Lines.

have

$$\begin{aligned}
1 &\leq \frac{\mathrm{EMD}_2(A, R_\phi(B))}{\mathrm{EMD}_2(A, B)}, \quad \text{since } A \text{ and } B \text{ are in optimal position} \\
&\leq \frac{\sum_{l\in L} \alpha_l \, \|l - R_\phi(b_l)\|_2 + \sum_{r\in R} \alpha_r \, \|r - R_\phi(b_r)\|_2}{\sum_{l\in L} \alpha_l \, \|l - b_l\|_2 + \sum_{r\in R} \alpha_r \, \|r - b_r\|_2}, \quad \text{by inequality (3.3)} \\
&\leq \frac{\sum_{l\in L} \alpha_l \, (\|l - l'\|_2 + \|l' - R_\phi(b_l)\|_2) + \sum_{r\in R} \alpha_r \, (\|r - r'\|_2 + \|r' - R_\phi(b_r)\|_2)}{\sum_{l\in L} \alpha_l \, \|l - b_l\|_2 + \sum_{r\in R} \alpha_r \, \|r - b_r\|_2} \\
&= \frac{\sum_{l\in L} \alpha_l \, (\|l - l'\|_2 + \|l' - R_\phi(b_l)\|_2) + \sum_{r\in R} \alpha_r \, (\|r - r'\|_2 + \|r' - R_\phi(b_r)\|_2)}{\sum_{l\in L} \alpha_l \, (\|l - l'\|_2 + \|l' - b_l\|_2) + \sum_{r\in R} \alpha_r \, (\|r - r'\|_2 + \|r' - b_r\|_2)} \\
&\leq \frac{\sum_{l\in L} \alpha_l \, \|l' - R_\phi(b_l)\|_2 + \sum_{r\in R} \alpha_r \, \|r' - R_\phi(b_r)\|_2}{\sum_{l\in L} \alpha_l \, \|l' - b_l\|_2 + \sum_{r\in R} \alpha_r \, \|r' - b_r\|_2}.
\end{aligned}$$

Thus, in the following we assume that all points in A lie either on C_r or C_l. We now show that we can assume that the points b_i equal one fixed point $b^* = (0, 1)$. To keep the system of forces in equilibrium and to leave the value of the upper bound unchanged, we scale the points a_i and b_i by $1/\|b_i\|_2$ and simultaneously scale their weights by the inverse factor $\|b_i\|_2$. See Figure 3.11 for an illustration of the resulting weighted point sets. The correctness of this step is an immediate consequence of the theorems of intersecting lines:

$$\begin{aligned}
\frac{\mathrm{EMD}_2(A, R_\phi(B))}{\mathrm{EMD}_2(A, B)} &\leq \frac{\sum_{i=1}^k \alpha_i \, \|a_i - R_\phi(b_i)\|_2}{\sum_{i=1}^k \alpha_i \, \|a_i - b_i\|_2} \\
&= \frac{\sum_{i=1}^k \alpha_i \, \|b_i\|_2 \|a_i' - R_\phi(b^*)\|_2}{\sum_{i=1}^k \alpha_i \, \|b_i\|_2 \|a_i' - b^*\|_2} \\
&= \frac{\sum_{i=1}^k \alpha_i' \, \|a_i' - R_\phi(b^*)\|_2}{\sum_{i=1}^k \alpha_i' \, \|a_i' - b^*\|_2} \qquad (3.5)
\end{aligned}$$

Thus for the following we assume that the points $b_1, \ldots, b_k$ coincide and we are in the situation of Figure 3.11. Above we have proved that

$$\sum_{\{i\,:\,(a_i)_x>0\}} \alpha_i \, \|b_i\|_2 \cos\gamma_i \;-\; \sum_{\{i\,:\,(a_i)_x<0\}} \alpha_i \, \|b_i\|_2 \cos\gamma_i \;=\; 0,$$

see formula (3.4). It is easy to observe that this formula still holds for the upper bound (3.5). Since here $\|b_i\|_2 = \|b^*\|_2$ for all i, we have

$$\sum_{l\in L} \alpha_l \cos\gamma_l \;=\; \sum_{r\in R} \alpha_r \cos\gamma_r.$$

Thus, by eventually subdividing points we can assume that A and B consist of pairs of points $(l_i, r_i)_{i=1,\ldots,k}$, such that $\alpha_{l_i}\cos\gamma_{l_i} = \alpha_{r_i}\cos\gamma_{r_i}$. Therefore,

$$\begin{aligned}
(3.5) \;&=\; \frac{\sum_{i=1}^k \alpha_{l_i}\,\|l_i - R_\phi(b^*)\|_2 + \sum_{i=1}^k \alpha_{r_i}\,\|r_i - R_\phi(b^*)\|_2}{\sum_{i=1}^k \alpha_{l_i}\,\|l_i - b^*\|_2 + \sum_{i=1}^k \alpha_{r_i}\,\|r_i - b^*\|_2} \\
&=\; \frac{\sum_{i=1}^k (\alpha_{l_i}\,\|l_i - R_\phi(b^*)\|_2 + \alpha_{r_i}\,\|r_i - R_\phi(b^*)\|_2)}{\sum_{i=1}^k (\alpha_{l_i}\,\|l_i - b^*\|_2 + \alpha_{r_i}\,\|r_i - b^*\|_2)} \\
&\le\; \sup_{i=1,\ldots,n} \frac{\alpha_{l_i}\,\|l_i - R_\phi(b^*)\|_2 + \alpha_{r_i}\,\|r_i - R_\phi(b^*)\|_2}{\alpha_{l_i}\,\|l_i - b^*\|_2 + \alpha_{r_i}\,\|r_i - b^*\|_2} \\
&=\; \sup g(l, r, \phi),
\end{aligned}$$

where

$$g(l, r, \phi) \;:=\; \frac{\alpha_l\,\|l - R_\phi(b^*)\|_2 + \alpha_r\,\|r - R_\phi(b^*)\|_2}{\alpha_l\,\|l - b^*\|_2 + \alpha_r\,\|r - b^*\|_2}.$$

A bound for $g(l, r, \phi)$ will immediately give a bound for (3.5). We bound this supremum in the following way: Let $(r, l) \in C_r \setminus \{0\} \times C_l \setminus \{0\}$. Using $\alpha_l \cos\gamma_l = \alpha_r \cos\gamma_r$ we have

$$g(l, r, \phi) \;=\; \frac{\cos\gamma_r\,\|l - R_\phi(b^*)\|_2 + \cos\gamma_l\,\|r - R_\phi(b^*)\|_2}{\cos\gamma_r\,\|l - b^*\|_2 + \cos\gamma_l\,\|r - b^*\|_2}.$$

Now, g is a function in the 3 unknowns l, r and ϕ, where, with a slight abuse of notation, l and r denote the distance of the points l and r to the origin. In Figure 3.12 we have illustrated g, where we fixed l to 0.99. The variables r and ϕ vary, such that $0.98 \le r \le 1.02$ and $0.01 \le \sin\phi \le 0.02$.

We now use MAPLE to maximize g for fixed values of l. We do this in the following way: We fix some value $l \in \mathbb{R}_{>0}$, $l \ne 1$, let $g_l(r, \phi)$ be this function for a fixed value of l. We then compute the partial derivatives $\partial g_l/\partial r$ and $\partial g_l/\partial\phi$. We then solve the system of equations

$$\frac{\partial g_l}{\partial r}(r, \phi) \;=\; 0 \quad\text{and}\quad \frac{\partial g_l}{\partial \phi}(r, \phi) \;=\; 0.$$

For every value of l this system of equations has exactly 3 solutions. Note that we solve the system of equations algebraically and therefore there cannot be more than this 3 candidates for extrema of g_l. One of the solutions is always assumed for $r = 1$ and is always a minimum of g_l. The other two solutions describe maxima of g_l. Let (r_1^*, ϕ_1^*) and (r_2^*, ϕ_2^*) be the maxima. We list a few important properties:

1. $r_1^* = 1/r_2^*$

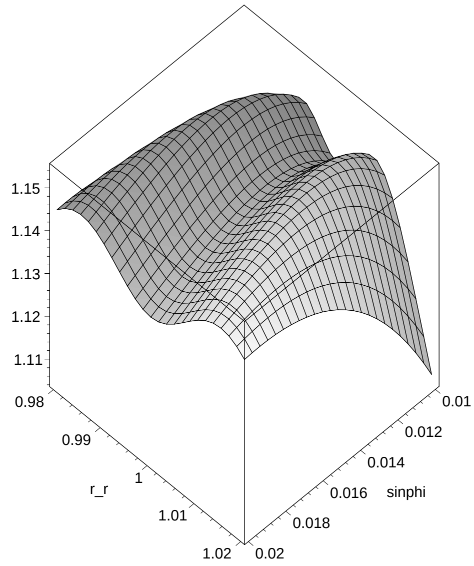


Figure 3.12: The function g near the maxima for $l = 0.99$.

2. $\phi_1^* = \phi_2^*$
3. $g_l(r_1^*, \phi_1^*) = g_l(r_2^*, \phi_2^*)$

In Appendix A we show a table of the results and Figure 3.13 depicts the resulting values $g_l(r_1^*, \phi_1^*)$ for different values $0 < l \leq 2$.

Observing this function and the list of values given in Appendix A, the result strongly recommends an upper bound smaller than 1.155 for the approximation ratio for the EMD_2 under rotations. Unfortunately, a formal proof of this has to be postponed to future work. One approach may be to bound $\partial g / \partial l$ by a constant. Then, by eventually increasing the number of points l near 1, where we compute the maxima of g, we can prove a good upper bound.

Approximation Algorithms for Rigid Motions in the Plane

Based on Lemma 3.4 we construct an approximation algorithm for minimizing the EMD_2 under rotations, which we can use to construct an approximation algorithm for rigid motions. In this section we discuss the planar case. The general case will be addressed in the following paragraph.

Consider the following method to find an approximation on the minimum EMD_2 under rotations around the fixed rotation center $p^* \in \mathbb{R}^d$.

Algorithm 3.5.

1. Determine a rotation $R' \in \mathrm{Rot}(p^*)$ minimizing $\mathrm{EMD}_2(A, R'(B))$ over all possible alignments of p^*, a point of A, and a point of B.

By Lemma 3.4 it follows that $\mathrm{EMD}_2(A, R'(B)) \leq 2 \cdot \min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_2(A, R(B))$. There

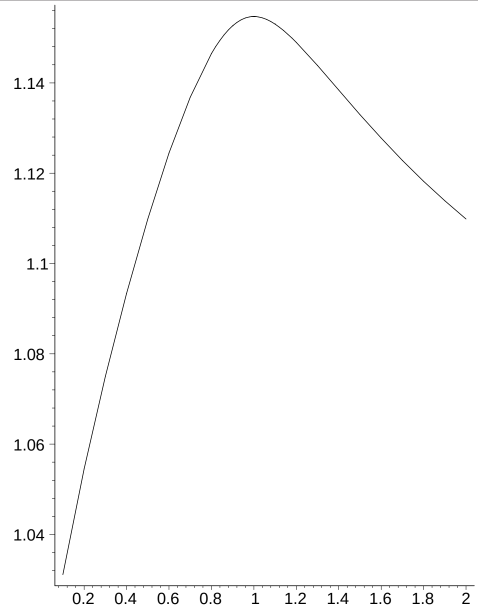


Figure 3.13: g maximized for varying l.

are $O(nm)$ possibilities to align p^* and any two points of A and B. Hence the runtime of this algorithm is $O(nm \cdot T^{\mathrm{EMD}_2}(n,m))$. We combine this algorithm with the abstract Algorithm 1.2 for rigid motions, see Section 1.5.3.

Algorithm 3.6.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.
2. Find a best matching of A and B' under rotations of B' around $r(A)$, where $r(A)$, a point of A, and a point of B' are aligned.
   Let B'' be the image of B' under this rotation.
3. Output B'' together with the distance $\mathrm{EMD}_2(A, B'')$.

Using the above remarks and Theorem 3.1, we can use our abstract algorithm for rigid motions, see Theorem 1.7, to prove the following result:

Theorem 3.13. *Let $A, B \in \mathbb{W}^{2,G}$ for some $G \in \mathbb{R}_{>0}$ be two planar weighted point sets. Let $r\colon \mathbb{W}^{2,G} \to \mathbb{R}^2$ be an* EMD_2*-reference point with respect to rigid motions and with Lipschitz constant c. Algorithm 3.6 finds an approximately optimal matching for* EMD_2 *under rigid motions with approximation factor* $2(c+1)$ *in time* $O(T^{\mathrm{ref}}(\max\{n,m\}) + nm \cdot T^{\mathrm{EMD}_2}(n,m))$.

Substituting EMD_2 by EMD_p in Algorithm 3.6 we obtain a constant-factor approximation for the EMD_p under rigid motions. Unfortunately, the approximation ratio is slightly worse. Again we prove the result using our abstract algorithm for rigid motions, applying Lemma 3.5 instead of Lemma 3.4.

Theorem 3.14. *Let $1 \leq p \leq \infty$. Let $r\colon \mathbb{W}^{2,G} \to \mathbb{R}^2$ be an EMD_p-reference point with respect to rigid motions and with Lipschitz constant c. Algorithm 3.6 finds an approximately optimal matching for EMD_p under rigid motions in the plane with approximation factor $2\sqrt{2}\,(c+1)$ in time $O(T^{\mathrm{ref}}(\max\{n,m\}) + nm \cdot T^{\mathrm{EMD}_p}(n,m))$.*

In the following corollary we apply the center of mass to the last two theorems:

Corollary 3.4. *For weighted point sets in the plane, Algorithm 3.6 using the center of mass as an EMD-reference point induces an approximation algorithm with approximation factor 4 in the Euclidean case and $4\sqrt{2}$ for any other L_p-distance, where $1 \leq p \leq \infty$. Its runtime is $O(nm \cdot T^{\mathrm{EMD}_p}(n,m))$.*

Approximation Algorithms for Rigid Motions in Higher Dimensions.

For the approximations in higher dimensions we need to have a higher dimensional transformation similar to the transformation of aligning two points and a rotation center in the plane. For affine spaces $S_1, \ldots, S_n$ we use $\mathrm{aff}(S_1, \ldots, S_n)$ to denote the affine space spanned by $S_1, \ldots, S_n$. Using this notation we give the following definition:

Definition 3.5. Let S^* be a d'-dimensional affine subspace in $\mathbb{R}^d$, where $0 \leq d' < d-1$. Let $a, b \in \mathbb{R}^d \setminus S^*$. Let $S_a := \mathrm{aff}(S^*, a)$ be the $(d'+1)$-dimensional affine space spanned by S^* and a. Note that $S_a \setminus S^*$ has exactly two connected disjoint components, where a lies in one of them. We say a and b are aligned with respect to S^*, if $\dim \mathrm{aff}(S^*, a, b) = d'+1$ and b lies in the same component as a. We also say b lies on the same side of S^* as a.

We use the following corollary of Lemmas 3.4 and 3.6 to obtain an approximation algorithm for the EMD under rigid motions in arbitrary dimension $d \geq 3$.

Corollary 3.5. *Let A, B be two weighted point sets in dimension $d \geq 3$. Let A, B be full-dimensional, that is $\dim A = \dim B = d$. Let $p^* \in \mathbb{R}^d$ be an arbitrary point. There are sequences of points $a^{(1)}, \ldots, a^{(d-1)} \in A$ and $b^{(1)}, \ldots, b^{(d-1)} \in B$, and a sequence of rotations $R^{(1)}, \ldots, R^{(d-1)} \in \mathrm{Rot}(p^*)$, such that the following conditions hold for any $1 \leq k \leq d-1$:*

1. *$S^{(0)} = p^*$ and for $k > 0$,*

$$S^{(k)} \;=\; \mathrm{aff}\left(p^*, a^{(1)}, \ldots, a^{(k)}, R^{(k)} \circ \cdots \circ R^{(1)}(b^{(1)}), \ldots, R^{(k)} \circ \cdots \circ R^{(1)}(b^{(k)})\right).$$

2. *$R^{(k)}$ leaves $S^{(k-1)}$ fixed,*

3. *$R^{(k)}$ aligns $a^{(k)}$ and $b^{(k)}$ with respect to $S^{(k-1)}$,*

4. *$\mathrm{EMD}_2(A, R^{(k)} \circ \cdots \circ R^{(1)}(B)) \;\leq\; 2^k \cdot \min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_2(A, R(B))$,*

Proof. We prove the result by induction on k. For $k = 1$ we can apply Lemma 3.4. Let $1 \leq k < d-1$. By induction we have $\dim S^{(k)} = k$ and

$$\mathrm{EMD}_2(A, R^{(k)} \circ \cdots \circ R^{(1)}(B)) \;\leq\; 2^k \cdot \min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_2(A, R(B)).$$

Let $R^* := \arg\min_{R \in \mathrm{Rot}(S^{(k)})} \mathrm{EMD}_2(A, R \circ R^{(k)} \circ \cdots \circ R^{(1)}(B))$. Of course,

$$\begin{aligned} \mathrm{EMD}_2(A, R^* \circ R^{(k)} \circ \cdots \circ R^{(1)}(B)) &\leq \mathrm{EMD}_2(A, R^{(k)} \circ \cdots \circ R^{(1)}(B)) \\ &\leq 2^k \cdot \min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_2(A, R(B)). \end{aligned}$$

Since A and B are full-dimensional we can apply Lemma 3.6 to find a pair of points $a^{(k+1)}$ and $R^{(k)} \circ \cdots \circ R^{(1)}(b^{(k+1)})$, together with a rotation $R^{(k+1)} \in \mathrm{Rot}(S^{(k)})$, such that $a^{(k+1)}$ and $R^{(k+1)} \circ \cdots \circ R^{(1)}(b^{(k+1)})$ are aligned with respect to $S^{(k)}$ and

$$\begin{aligned}\mathrm{EMD}_2(A, R^{(k+1)} \circ \cdots \circ R^{(1)}(B)) \;&\leq\; 2 \cdot \mathrm{EMD}_2(A, R^* \circ R^{(k)} \circ \cdots \circ R^{(1)}(B)) \\ &\leq\; 2^{k+1} \cdot \min_{R \in \mathrm{Rot}(p^*)} \mathrm{EMD}_2(A, R(B)).\end{aligned}$$

□

For the next theorem we need to show that the rotation defined in the last lemma is unique:

Lemma 3.7. *Let $p^* \in \mathbb{R}^d$ be a point and let $R \in \mathrm{Rot}(p^*)$ be a rotation. Let $a^{(1)}, \ldots, a^{(d-1)}$ and $b^{(1)}, \ldots, b^{(d-1)}$ be two sequences of points in $\mathbb{R}^d$ such that $\mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(d-1)})$ and $\mathrm{aff}(p^*, b^{(1)}, \ldots, b^{(d-1)})$ are affinely independent. Further, for every $1 \leq k \leq d-1$ let $a^{(k)}$ and $R(b^{(k)})$ be aligned with respect to $\mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(k-1)})$. Then,*

1. *for all $0 \leq k \leq d-1$ we have $\mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(k)}) = R\left(\mathrm{aff}(p^*, b^{(1)}, \ldots, b^{(k)})\right)$,*
2. *and R is uniquely determined.*

Proof. We first prove claim 1 and the fact that the positions of $R(b^{(1)}), \ldots, R(b^{(d-1)})$ are uniquely determined. We prove this by induction: For $k = 1$, R rotates $b^{(1)}$ around p^* such that $R(b^{(1)})$ lies on the ray with origin p^* in direction $a^{(1)}$. This position of $R(b^{(1)})$ is unique. We further have that $\mathrm{aff}(p^*, a^{(1)}) = R\left(\mathrm{aff}(p^*, b^{(1)})\right)$.

Let the result be true for some $1 \leq k < d-1$, that is, the positions of $R(b^{(1)}), \ldots, R(b^{(k)})$ are uniquely determined and claim 1 holds for k.

We know, $a^{(k+1)}$ and $R(b^{(k+1)})$ are aligned with respect to $\mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(k)})$. Thus, $R(b^{(k+1)})$ lies in the $(k+1)$-dimensional space X_1 spanned by $\mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(k+1)})$. Thus, the possible positions of $R(b^{(k+1)})$ lie on a $(d-k)$-dimensional affine space X_2 orthogonal to $R(\mathrm{aff}(p^*, b^{(1)}, \ldots, b^{(k)})) = \mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(k)})$. Therefore $R(b^{(k+1)})$ lies in $X_1 \cap X_2$, which is a 1-dimensional space orthogonal to $\mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(k)})$. Further, the distance of $R(b^{(k+1)})$ to $\mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(k)})$ is the distance from $b^{(k+1)}$ to $\mathrm{aff}(p^*, b^{(1)}, \ldots, b^{(k)})$, and with this distance there are 2 possible points on the line $X_1 \cap X_2$. One of these points lies on the same side of $\mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(k)})$ as $a^{(k+1)}$ and one on the other. Since $R(b^{(k+1)})$ lies on the same side, the position of this point is uniquely determined. We further have that $\mathrm{aff}(p^*, a^{(1)}, \ldots, a^{(k+1)}) = R(\mathrm{aff}(p^*, b^{(1)}, \ldots, b^{(k+1)}))$.

By induction we know that the images of d affinely independent points $p^*, R^{(1)}, \ldots, R^{(d-1)}$ under the rotation R are uniquely determined. By basic linear algebra we know that R is uniquely determined. □

Based on the abstract Algorithm 1.2 in Section 1.5.3 we get an approximation algorithm for minimizing the EMD_2 under rigid motions.

Theorem 3.15. *Let $A, B \in \mathbb{W}^{d,G}$, where $d \geq 3$ and $G \in \mathbb{R}_{>0}$. Let A, B be full-dimensional, that is $\dim A = \dim B = d$. Let $r \colon \mathbb{W}^{d,G} \to \mathbb{R}^d$ be an EMD_2-reference point with respect to rigid motions and with Lipschitz constant c. An approximate rigid motion M', such that*

$$\mathrm{EMD}_2(A, M'(B)) \;\leq\; 2^{d-1}(c+1) \cdot {\mathrm{EMD}_2}^{\mathrm{opt}}(A, B)$$

can be found in $O(T^{\mathrm{ref}}(\max\{n, m\}) + n^{d-1} m^{d-1} \cdot T^{\mathrm{EMD}_2}(n, m))$ time.

Proof. As in the planar case we first translate B such that the reference points of A and B coincide. Algorithmically we proceed in the following way: We choose a pair of points $(a^{(1)}, b^{(1)})$ in $A \times B$. We rotate B around the reference point such that $a^{(1)}, b^{(1)}$ and the reference point are aligned. We fix the line determined by these points. We then choose a second pair of points $(a^{(2)}, b^{(2)})$ and rotate B around the fixed line such that $b^{(2)}$ lies in the plane spanned by $a^{(2)}$ and the fixed line. We fix the plane, choose another pair of points and continue until a $(d-1)$-dimensional subspace is fixed. We compute the EMD_2 between A and the rotated version of B and continue.

Altogether we compute the EMD_2 for every sequence of pairs of points as described in Corollary 3.5. Thus we have surely considered those sequences $a^{(1)}, \ldots, a^{(d-1)}$ and $b^{(1)}, \ldots, b^{(d-1)}$ and the corresponding rotation $R := R^{(d-1)} \circ \cdots \circ R^{(1)}$ that induce the approximation property in Corollary 3.5. Since by Lemma 3.7 the rotation R is unique, it equals the rotation constructed above and therefore we have proven the approximation property of the theorem.

Observing that there are $O(n^{d-1} \cdot m^{d-1})$ possibilities for these sequences proves the runtime of the algorithm. □

Note that if $d' := \dim B < \dim A$ it is enough to consider every sequence of d' pairs of points. After this, every rotation leaves B invariant and thus does not change the EMD_2 between the two sets.

Analogous to the proof of Lemma 3.5 we can extend the result to the EMD defined on any L_p-norm:

Theorem 3.16. *Let $A, B \in \mathbb{W}^{d,G}$, where $d \geq 3$ and $G \in \mathbb{R}_{>0}$, and let $1 \leq p \leq \infty$. Let A, B be full-dimensional, that is $\dim A = \dim B = d$. Let $r \colon \mathbb{W}^{d,G} \to \mathbb{R}^d$ be an EMD_p-reference point with respect to rigid motions and with Lipschitz constant c. An approximate rigid motion M' such that*

$$\mathrm{EMD}_p(A, M'(B)) \;\leq\; 2^{d-1}\sqrt{d}\,(c+1) \cdot \mathrm{EMD}_p^{\mathrm{opt}}(A, B)$$

can be found in $O(T^{\mathrm{ref}}(\max\{n, m\}) + n^{d-1}m^{d-1} \cdot T^{\mathrm{EMD}}(n, m))$ time.

We apply the center of mass and obtain the following corollary:

Corollary 3.6. *Applying the center of mass as an* EMD*-reference point to the algorithm described above, the method induces an approximation algorithm with approximation factor 2^d in the case of the Euclidean norm and $2^d\sqrt{d}$ for any other L_p-norm, where $1 \leq p \leq \infty$. Its runtime is $O(n^{d-1}m^{d-1} \cdot T^{\mathrm{EMD}_p}(n, m))$.*

3.4.5 Similarities

In this section we present approximation algorithms for matching weighted point sets under similarities, i.e., combinations of translations, rotations and scalings. More precisely, we want to compute $\min_S \mathrm{EMD}(A, S(B))$, where the minimum is taken over all similarities S. Note that in this case exchanging A and B makes a difference. We only consider positive similarities, i.e., the scaling factor is positive. The consideration of negative similarities is easy by using the same algorithms on B and a reflected copy of A.

Basically, the approach is to use the algorithm for rigid motions on the two weighted point sets, where B is scaled by the quotient of the normalized first moments with respect to their reference points. See also the comments in Section 1.5.4 on the abstract approximation algorithm for similarities.

We give the well-known definition of the normalized first moment of a weighted point set with respect to an arbitrary point $p^* \in \mathbb{R}^d$.

Definition 3.6 (Normalized First Moment)**.** Let $A = \{(p_i, \alpha_i)_{i=1,\dots,n}\} \in \mathbb{W}^{d,G}$ be a weighted point set for some $G \in \mathbb{R}_{>0}$, and let $p^* \in \mathbb{R}^d$ be an arbitrary point. We call

$$m_{p^*}(A) \;=\; \frac{1}{W^A} \sum_{i=1}^{n} \alpha_i \, \|p_i - p^*\|$$

the normalized first moment of A with respect to p^*.

Note that the normalized first moment of a weighted point set with respect to an arbitrary point can be calculated efficiently in linear time.

Consider the following algorithm to approximately compute the EMD under similarities:

Algorithm 3.7.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.
2. Determine the normalized first moments $m_{r(A)}(A)$ and $m_{r(B')}(B')$, and scale B' by $m_{r(A)}(A)/m_{r(B')}(B')$ around the center $r(A)$.
   Let B'' be the image of B' under this scaling.
3. Find an optimal matching of A, B'' under rotations of B'' around $r(A)$.
   Let B''' be the image of B'' under this rotation.
4. Output B''' together with the distance $\mathrm{EMD}(A, B''')$.

We now analyze the approximation factor of this algorithm. First we show that it is trivial to compute the EMD between two weighted point sets which are scalings of the same weighted point set.

Lemma 3.8. *Let $A \in \mathbb{W}^{d,G}$ for some $G \in \mathbb{R}_{>0}$ be a weighted point set, and let $m_{p^*}(A)$ be its normalized first moment with respect to some point $p^* \in \mathbb{R}^d$. Let τ_1, τ_2 be positive scalings around the same center p^* with ratios γ_1 and γ_2, respectively. Then*

$$\mathrm{EMD}\left(\tau_1(A), \tau_2(A)\right) \;=\; |\gamma_1 - \gamma_2| \cdot m_{p^*}(A).$$

Proof.

$$\begin{aligned}
\mathrm{EMD}\left(\tau_1(A), \tau_2(A)\right) \;&\leq\; \frac{1}{G} \sum_{i=1}^{n} \alpha_i \, \|\tau_1(p_i) - \tau_2(p_i)\| \\
&=\; \frac{1}{G} \sum_{i=1}^{n} \alpha_i \, \|p^* + \gamma_1(p_i - p^*) - (p^* + \gamma_2(p_i - p^*))\| \\
&=\; \frac{1}{G} \sum_{i=1}^{n} \alpha_i \, \|(\gamma_1 - \gamma_2)(p_i - p^*)\| \\
&=\; \frac{|\gamma_1 - \gamma_2|}{G} \sum_{i=1}^{n} \alpha_i \, \|p_i - p^*\| \\
&=\; |\gamma_1 - \gamma_2| \cdot m_{p^*}(A)
\end{aligned}$$

In the first inequality we have chosen the flow between the corresponding scaled points. This is a feasible flow and therefore the inequality holds.

We now prove that $\mathrm{EMD}(\tau_1(A), \tau_2(A)) \geq |\gamma_1 - \gamma_2| \cdot m_{p^*}(A)$. Thus we have to show that the flow variables described above are optimal. Let C be any cycle in the residual network. By the structure of the flow, C is always of the form

$$\tau_2(p^{(1)}) \to \tau_1(p^{(1)}) \to \tau_2(p^{(2)}) \to \tau_1(p^{(2)}) \to \cdots \to \tau_2(p^{(k)}) \to \tau_1(p^{(k)}) \to \tau_2(p^{(1)})$$

for a list of points $p^{(1)}, \ldots, p^{(k)} \in A$. W.l.o.g. let $\gamma_2 \geq \gamma_1$. Let $p^{(k+1)} := p^{(1)}$. Then, the residual costs of this cycle can be computed as

$$\begin{aligned}
&\sum_{i=1}^{k} \left(\|\tau_2(p^{(i+1)}) - \tau_1(p^{(i)})\| - \|\tau_1(p^{(i)}) - \tau_2(p^{(i)})\| \right) \\
&= \sum_{i=1}^{k} \left\| \tau_2(p^{(i+1)}) - p^* - (\tau_1(p^{(i)}) - p^*) \right\| - \sum_{i=1}^{k} \|\tau_1(p^{(i)}) - \tau_2(p^{(i)})\| \\
&\geq \sum_{i=1}^{k} \left| \|\tau_2(p^{(i+1)}) - p^*\| - \|\tau_1(p^{(i)}) - p^*\| \right| - \sum_{i=1}^{k} \|\tau_1(p^{(i)}) - \tau_2(p^{(i)})\| \\
&\geq \sum_{i=1}^{k} \left| \|\tau_2(p^{(i)}) - p^*\| - \|\tau_1(p^{(i)}) - p^*\| \right| - \sum_{i=1}^{k} \|\tau_1(p^{(i)}) - \tau_2(p^{(i)})\| \qquad (3.6) \\
&= \sum_{i=1}^{k} \|\tau_2(p^{(i)}) - \tau_1(p^{(i)})\| - \sum_{i=1}^{k} \|\tau_1(p^{(i)}) - \tau_2(p^{(i)})\| \\
&= 0.
\end{aligned}$$

It remains to show that inequality (3.6) holds. Thus we have to bound

$$\begin{aligned}
&\sum_{i=1}^{k} \left| \|\tau_2(p^{(i+1)}) - p^*\| - \|\tau_1(p^{(i)}) - p^*\| \right| \\
&= \sum_{i=1}^{k} \left| \gamma_2 \|p^{(i+1)} - p^*\| - \gamma_1 \|p^{(i)} - p^*\| \right|. \qquad (3.7)
\end{aligned}$$

Consider the two lists of points $V := v_1, \ldots, v_k$ and $W := w_1, \ldots, w_k$, where $v_i := \gamma_1 \|p^{(i)} - p^*\|$ and $w_i := \gamma_2 \|p^{(i)} - p^*\|$. Formula (3.7) describes the cost of a perfect matching between the points in V and W. Since $w_i = \gamma_2/\gamma_1 \, v_i$ we have $v_i \leq v_j$ is equivalent to $w_i \leq w_j$ for two indices $1 \leq i, j \leq k$. Thus, the minimum cost perfect matching is given by the matching where v_i is matched to w_i for every $1 \leq i \leq k$. This fact is a special case of the considerations in Section 3.4.1. Thus we have

$$\begin{aligned}
(3.7) &\geq \sum_{i=1}^{k} |w_i - v_i| \\
&= \sum_{i=1}^{k} \left| \gamma_2 \|p^{(i)} - p^*\| - \gamma_1 \|p^{(i)} - p^*\| \right| \\
&= \sum_{i=1}^{k} \left| \|\tau_2(p^{(i)}) - p^*\| - \|\tau_1(p^{(i)}) - p^*\| \right|,
\end{aligned}$$

which completes the proof of inequality (3.6).

Altogether we have shown that there is no cycle with negative costs in the residual network. By basic network flow theory we conclude that the flow is optimal. □

We use the following lemma to prove a new lower bound for the EMD.

Lemma 3.9. *Let $A, B \in \mathbb{W}^{d,G}$ for some $G \in \mathbb{R}_{>0}$. Let p^* and q^* be two fixed points in $\mathbb{R}^d$. Then*

$$|m_{p^*}(A) - m_{q^*}(B)| \;\leq\; \mathrm{EMD}(A,B) + \|p^* - q^*\|.$$

Proof. Let $F^* := \{f^*_{ij}\}$ be a flow defining $\mathrm{EMD}(A,B)$.

$$\begin{aligned}
|m_{p^*}(A) - m_{q^*}(B)| &= \left| \frac{1}{G}\sum_{i=1}^{n} \alpha_i \|p_i - p^*\| - \frac{1}{G}\sum_{j=1}^{m} \beta_j \|q_j - q^*\| \right| \\
&= \frac{1}{G}\left| \sum_{i=1}^{n}\sum_{j=1}^{m} f^*_{ij}\|p_i - p^*\| - \sum_{j=1}^{m}\sum_{i=1}^{n} f^*_{ij}\|q_j - q^*\| \right| \\
&= \frac{1}{G}\left| \sum_{i=1}^{n}\sum_{j=1}^{m} f^*_{ij}\left(\|p_i - p^*\| - \|q_j - q^*\|\right) \right| \\
&\leq \frac{1}{G}\sum_{i=1}^{n}\sum_{j=1}^{m} f^*_{ij}\,\big|\, \|p_i - p^*\| - \|q_j - q^*\| \,\big| \qquad (3.8)
\end{aligned}$$

Now,

$$\begin{aligned}
\|p_i - p^*\| &= \|p_i - q_j + q_j - q^* + q^* - p^*\| \\
&\leq \|p_i - q_j\| + \|q_j - q^*\| + \|q^* - p^*\| \qquad (3.9)
\end{aligned}$$

and analogously

$$\|q_j - q^*\| \;\leq\; \|p_i - q_j\| + \|p_i - p^*\| + \|q^* - p^*\|. \qquad (3.10)$$

By inequalities (3.9) and (3.10) we immediately see

$$\big|\|p_i - p^*\| - \|q_j - q^*\|\big| \;\leq\; \|p_i - q_j\| + \|q^* - p^*\|. \qquad (3.11)$$

Therefore it follows

$$\begin{aligned}
|m_{p^*}(A) - m_{q^*}(B)| &\leq \frac{1}{G}\sum_{i=1}^{n}\sum_{j=1}^{m} f^*_{ij}\,\big|\, \|p_i - p^*\| - \|q_j - q^*\| \,\big|, \quad \text{by inequality (3.8)} \\
&\leq \frac{1}{G}\sum_{i=1}^{n}\sum_{j=1}^{m} f^*_{ij}\left(\|p_i - q_j\| + \|q^* - p^*\|\right), \quad \text{by inequality (3.11)} \\
&= \frac{1}{G}\sum_{i=1}^{n}\sum_{j=1}^{m} f^*_{ij}\|p_i - q_j\| + \frac{1}{G}\sum_{i=1}^{n}\sum_{j=1}^{m} f^*_{ij}\|q^* - p^*\| \\
&= \mathrm{EMD}(A,B) + \|q^* - p^*\|.
\end{aligned}$$

□

The next theorem gives a lower bound for the EMD based on the first moments of the weighted point sets. It would be interesting to see how this lower bound compares with the known lower bounds in applications; see also the work on image retrieval by Cohen [20] and the recent work by Assent, Wenning and Seidl [10]. In the latter paper the authors investigate a multi-step approach for efficient query processing in large multimedia databases. They use the lower bounds to speed up their approach.

Theorem 3.17. *Let $A, B \in \mathbb{W}^{d,G}$ for some $G \in \mathbb{R}_{>0}$ and let $r\colon \mathbb{W}^{d,G} \to \mathbb{R}^d$ be an* EMD*-reference point with Lipschitz constant c. Then*

$$\left|m_{r(A)}(A) - m_{r(B)}(B)\right| \;\leq\; (1+c)\cdot \mathrm{EMD}(A,B).$$

Proof. By Lemma 3.9 we have

$$\left|m_{r(A)}(A) - m_{r(B)}(B)\right| \;\leq\; \mathrm{EMD}(A,B) + \|r(B) - r(A)\|.$$

The claim follows by Lipschitz continuity. □

Applying the center of mass as an EMD-reference point with Lipschitz constant 1 immediately proves the following corollary:

Corollary 3.7. *Let $A, B \in \mathbb{W}^{d,G}$ for some $G \in \mathbb{R}_{>0}$. Then*

$$\left|m_{C(A)}(A) - m_{C(B)}(B)\right| \;\leq\; 2\cdot \mathrm{EMD}(A,B).$$

Using these results we can prove that Algorithm 3.7 leads to a constant-factor approximation for minimizing the Euclidean EMD under similarities:

Theorem 3.18. *Let $r\colon \mathbb{W}^{d,G} \to \mathbb{R}^d$ be an* EMD_2*-reference point with respect to similarities and with Lipschitz constant c. Algorithm 3.7 finds an approximately optimal matching for similarities with approximation factor $2(c+1)$ in time $O(T^{\mathrm{ref}}(\max\{n,m\}) + T^{\mathrm{EMD}_2}(n,m) + T^{\mathrm{rot}}(n,m))$. This holds for arbitrary dimension $d \geq 2$.*

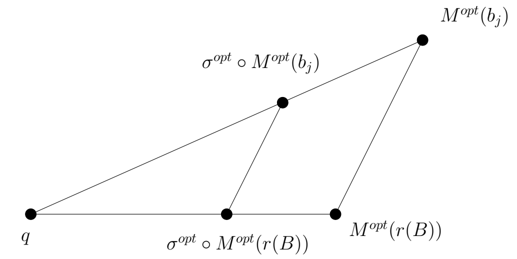
Proof. Consider an optimal similarity S^{opt}. It can be written as $S^{\mathrm{opt}} = \sigma^{\mathrm{opt}} \circ M^{\mathrm{opt}}$, where M^{opt} is a rigid motion and σ^{opt} is a scaling with ratio α^{opt} around some point q. Let $\delta := \mathrm{EMD}_2(A, S^{\mathrm{opt}}(B))$ be the optimal EMD_2 under similarities. Then,

$$\|r(A) - r(S^{\mathrm{opt}}(B))\|_2 \;\leq\; c\delta$$

because of the Lipschitz continuity of the EMD_2-reference point r. Let τ^r be the translation by $r(A) - r(S^{\mathrm{opt}}(B))$; then $\tilde{S} := \tau^r \circ S^{\mathrm{opt}}$ is a similarity mapping $r(B)$ onto $r(A)$ and

$$\begin{aligned}
\mathrm{EMD}_2(A, \tilde{S}(B)) &= \mathrm{EMD}_2(A, \tau^r \circ S^{\mathrm{opt}}(B)) \\
&\leq \mathrm{EMD}_2(A, S^{\mathrm{opt}}(B)) + \mathrm{EMD}_2(S^{\mathrm{opt}}(B), \tau^r \circ S^{\mathrm{opt}}(B)) \\
&= \delta + \|\tau^r\|_2, \quad \text{by Theorem 3.1} \\
&\leq \delta + c\delta.
\end{aligned}$$

Now write $\tilde{S}$ as $\tilde{S} = \tilde{\sigma} \circ \tilde{M}$, where $\tilde{M}$ is a rigid motion mapping $r(B)$ onto $r(A)$ and $\tilde{\sigma}$ is a scaling with center $r(A)$ and ratio α^{opt}. Let $\alpha := m_{r(A)}(A)/m_{r(B)}(B)$, σ the scaling with



$$\frac{\|\sigma^{opt} \circ M^{opt}(b_j) - \sigma^{opt} \circ M^{opt}(r(B))\|_2}{\|M^{opt}(b_j) - M^{opt}(r(B))\|_2} = \frac{\|\sigma^{opt} \circ M^{opt}(r(B)) - q\|_2}{\|M^{opt}(r(B)) - q\|_2} = \alpha^{opt}$$

Figure 3.14: Using the Theorem of Intersecting Lines.

center $r(A)$ and ratio α, and $S = \sigma \circ \tilde{M}$. Then,

$$\begin{aligned}
\mathrm{EMD}_2\left(\tilde{S}(B), S(B)\right) &= \mathrm{EMD}_2\left(\tilde{\sigma} \circ \tilde{M}(B), \sigma \circ \tilde{M}(B)\right) \\
&= \left|(\alpha^{\mathrm{opt}} - \alpha) \cdot m_{r(A)}(\tilde{M}(B))\right|, \quad \text{by Lemma 3.8} \\
&= \left|(\alpha^{\mathrm{opt}} - \alpha) \cdot m_{r(B)}(B)\right| \qquad (3.12) \\
&= \left|\alpha^{\mathrm{opt}} \cdot m_{r(B)}(B) - \alpha \cdot m_{r(B)}(B)\right| \\
&= \left|\alpha^{\mathrm{opt}} \cdot m_{r(B)}(B) - m_{r(A)}(A)\right|, \quad \text{by definition of } \alpha \\
&= \left|m_{r(S^{\mathrm{opt}}(B))}(S^{\mathrm{opt}}(B)) - m_{r(A)}(A)\right| \qquad (3.13) \\
&\leq (1+c) \cdot \mathrm{EMD}_2(S^{\mathrm{opt}}(B), A), \quad \text{by Theorem 3.17.}
\end{aligned}$$

It remains to show that equations (3.12) and (3.13) hold:

- For equation (3.12) we have to show that $m_{r(A)}(\tilde{M}(B)) = m_{r(B)}(B)$. This holds because $\tilde{M}$ is a rigid motion mapping $r(B)$ onto $r(A)$ and a rigid motion does not affect the distances to a point which is translated and rotated in the same way.
- For equation (3.13) we have to show that $\alpha^{\mathrm{opt}} m_{r(B)}(B) = m_{r(S^{\mathrm{opt}}(B))}(S^{\mathrm{opt}}(B))$.

$$\begin{aligned}
m_{r(S^{\mathrm{opt}}(B))}\left(S^{\mathrm{opt}}(B)\right) &= m_{r(\sigma^{\mathrm{opt}} \circ M^{\mathrm{opt}}(B))}\left(\sigma^{\mathrm{opt}} \circ M^{\mathrm{opt}}(B)\right) \\
&= m_{\sigma^{\mathrm{opt}} \circ M^{\mathrm{opt}}(r(B))}\left(\sigma^{\mathrm{opt}} \circ M^{\mathrm{opt}}(B)\right) \\
&= \frac{1}{G} \sum_{j=1}^{m} \beta_j \left\|\sigma^{\mathrm{opt}} \circ M^{\mathrm{opt}}(b_j) - \sigma^{\mathrm{opt}} \circ M^{\mathrm{opt}}(r(B))\right\|_2 \\
&= \frac{1}{G} \sum_{j=1}^{m} \beta_j \, \alpha^{\mathrm{opt}} \left\|M^{\mathrm{opt}}(b_j) - M^{\mathrm{opt}}(r(B))\right\|_2, \quad \text{Fig. 3.14} \\
&= \alpha^{\mathrm{opt}} \cdot m_{M^{\mathrm{opt}}(r(B))}(M^{\mathrm{opt}}(B)) \\
&= \alpha^{\mathrm{opt}} \cdot m_{r(B)}(B)
\end{aligned}$$

Altogether we have

$$\mathrm{EMD}_2(A, S(B)) \;\leq\; \mathrm{EMD}_2(A, \tilde{S}(B)) + \mathrm{EMD}_2(\tilde{S}(B), S(B)) \;\leq\; 2(c+1)\delta$$

for some similarity S composed of a rigid motion that maps $r(B)$ onto $r(A)$ and a scaling with center $r(A)$ and ratio α. Since Algorithm 3.7 finds the optimum among these similarities, the bound holds for it as well. The runtime of this algorithm depends on the time to compute (i) the EMD_2-reference points, (ii) translate B such that the EMD_2-reference points coincide, (iii) scale the translated version of B, (iv) find the optimal rotation around $r(A)$ and (v) compute the EMD_2 between A and the transformed version of B. Since computing the normalized first moment and therefore the scaling can be done in linear time, the time bound remains the same as for Algorithm 3.4. □

We can generalize the last theorem to any L_p-distance, where $1 \le p \le \infty$. The proof is analogous to the proof of Theorem 3.5 and omitted here.

Theorem 3.19. *Let $1 \le p \le \infty$. Let $r\colon \mathbb{W}^{d,G} \to \mathbb{R}^d$ be an EMD_p-reference point with respect to similarities and with Lipschitz constant c. Algorithm 3.7 finds an approximately optimal matching for similarities with approximation factor $2\sqrt{d}\,(c+1)$ in time $O(T^{\text{ref}}(\max\{n,m\}) + T^{\text{EMD}}(n,m) + T^{\text{rot}}(n,m))$. This holds for arbitrary dimension $d \ge 2$.*

We apply the center of mass to obtain implementable algorithms.

Corollary 3.8. *Algorithm 3.7 using the center of mass as an EMD-reference point induces an approximation algorithm with approximation factor 4 in the Euclidean case and $4\sqrt{d}$ for any other L_p-distance, where $1 \le p \le \infty$. The runtime is $O(T^{\text{EMD}}(n,m) + T^{\text{rot}}(n,m))$. This holds for any dimension $d \ge 2$.*

Approximation Algorithm for Similarities in the Plane.

As in the case of Algorithm 3.4, Algorithm 3.7 depends on finding the optimal rotation, which is impractical. Again we make this algorithm practical and efficient by using the approximation algorithm for rotations and again we have to pay by a worse approximation factor. As before we start with the Euclidean EMD in the plane.

Algorithm 3.8.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.
2. Determine the normalized first moments $m_{r(A)}(A)$ and $m_{r(B')}(B')$, and scale B' by $m_{r(A)}(A)/m_{r(B')}(B')$ around $r(A)$.
   Let B'' be the image of B' under this scaling.
3. Find a best matching of A and B'' under rotations of B'' around $r(A)$, where $r(A)$, a point of A, and a point of B'' are aligned.
   Let B''' be the image of B'' under this rotation.
4. Output B''' together with the distance $\text{EMD}_2(A, B''')$.

Theorem 3.20. *Let $r\colon \mathbb{W}^{2,G} \to \mathbb{R}^2$ be an EMD_2-reference point for planar weighted point sets with respect to similarities and with Lipschitz constant c. Algorithm 3.8 finds an approximately optimal matching for similarities with approximation factor $4(c{+}1)$ in time $O(T^{\text{ref}}(\max\{n,m\}) + nm \cdot T^{\text{EMD}_2}(n,m))$.*

Proof. Let $A, B \in \mathbb{W}^{2,G}$ for some $G \in \mathbb{R}_{>0}$ be two planar weighted point sets. Let $\tau := r(A) - r(B)$, and let σ be the scaling by $m_{r(A)}(A)/m_{r(A)}(\tau(B))$. Let M^* be the rigid motion minimizing $\mathrm{EMD}_2(A, M \circ \sigma \circ \tau(B))$ while mapping $r(B)$ onto $r(A)$. Let M^{**} be the rigid motion minimizing $\mathrm{EMD}_2(A, M \circ \sigma \circ \tau(B))$ while mapping $r(B)$ onto $r(A)$ and additionally aligning $r(A)$ a point of A, and a point of B. Let $S^* = M^{**} \circ \sigma \circ \tau(B)$. Note that S^* is the similarity found by Algorithm 3.8. Then

$$\begin{aligned}\mathrm{EMD}_2\left(A, S^*(B)\right) &= \mathrm{EMD}_2\left(A, M^{**} \circ \sigma \circ \tau(B)\right) \\ &\leq 2 \cdot \mathrm{EMD}_2\left(A, M^* \circ \sigma \circ \tau(B)\right), \quad \text{by Lemma 3.4} \\ &\leq 4\,(1+c)\,\delta, \quad \text{by Theorem 3.18.}\end{aligned}$$

The runtime of this algorithm depends on the time to compute the EMD_2-reference points, translate B such that the EMD_2-reference points coincide and compute the EMD_2 at all $O(nm)$ possible alignments of points in A, B and $r(A)$. □

We extend the result to any L_p-distance, where $1 \leq p \leq \infty$. The proof is analogous to the proof of Theorem 3.5.

Theorem 3.21. *Regarding* EMD_p *in the plane, where* $1 \leq p \leq \infty$*, Algorithm 3.8 finds an approximately optimal matching for similarities with approximation factor* $4\sqrt{2}(c+1)$ *in time* $O(T^{\mathrm{ref}}(\max\{n,m\}) + nm \cdot T^{\mathrm{EMD}_p}(n,m))$*.*

Applying the center of mass to the last two theorems leads to the following corollary:

Corollary 3.9. *Algorithm 3.8 using the center of mass as an* EMD*-reference point induces an approximation algorithm with approximation factor* 8 *in the case of the Euclidean distance in the plane and* $8\sqrt{2}$ *for any other* L_p*-distance,* $1 \leq p \leq \infty$*. Its runtime is* $O(nm \cdot T^{\mathrm{EMD}_p}(n,m))$*.*

Similarities in Higher Dimensions.

As in the corresponding Section 3.4.4 for rigid motions, we can use Algorithm 3.8 to construct an algorithm for dimensions ≥ 3. Working out the details in Section 3.4.4, one sees that the necessary changes only concern the process of finding an approximate rotation. Therefore we can use the same approach and obtain the following results:

Theorem 3.22. *Let* $A, B \in \mathbb{W}^{d,G}$*, where* $d \geq 3$*,* $G \in \mathbb{R}_{>0}$ *and* A, B *are full-dimensional. An approximate similarity* S' *such that*

$$\mathrm{EMD}_2(A, S'(B)) \;\leq\; 2^d(c+1) \cdot \mathrm{EMD}_2{}^{\mathrm{opt}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(\max\{n,m\}) + n^{d-1}m^{d-1} \cdot T^{\mathrm{EMD}}(n,m))$ *time.*

Again we can extend the algorithm to any L_p-distances:

Theorem 3.23. *Let* $A, B \in \mathbb{W}^{d,G}$*, where* $d \geq 3$*,* $G \in \mathbb{R}_{>0}$ *and* A, B *are full-dimensional. Let* $1 \leq p \leq \infty$*. An approximate rigid motion* S' *such that*

$$\mathrm{EMD}_p(A, S'(B)) \;\leq\; 2^d\sqrt{d}\,(c+1) \cdot \mathrm{EMD}_p^{\mathrm{opt}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(\max\{n,m\}) + n^{d-1}m^{d-1} \cdot T^{\mathrm{EMD}}(n,m))$ *time.*

We apply the center of mass to obtain implementable algorithms:

Corollary 3.10. *Applying the center of mass as an* EMD*-reference point to the algorithm described above, the method induces an approximation algorithm with approximation factor* 2^{d+1} *in the case of the Euclidean distance and* $2^{d+1}\sqrt{d}$ *for any other* L_p*-distance, where* $1 \leq p \leq \infty$*. Its runtime is* $O(n^{d-1}m^{d-1} \cdot T^{\mathrm{EMD}_p}(n,m))$.

3.4.6 Scalings that Do Not Work

In this section we show that scaling with respect to the diameter and the normalized second moment does not lead to a constant-factor approximation algorithm.

Diameter Does Not Work

In the last section we used the quotient of the normalized first moments of the two sets as approximate scaling factor. Intuitively, a more natural idea is to choose the quotient of the diameters of the sets. Here we show that this does not lead to a constant-factor approximation. The reason is that points of very small weight can determine the diameter, without having a big effect on the EMD. We show that this does not work for the center of mass as an EMD-reference point and therefore cannot work in general. We concentrate on the Euclidean case. It is easy to see that the result holds for any L_p-distance as well, where $1 \leq p \leq \infty$.

Consider the following two sets, where $K \geq 1$ is any positive constant:

$$\begin{aligned} A &:= \{((-1,0),1)\,,((1,0),1)\,,((0,0),2K-2)\}\,, \\ B &:= \{((-1,0),K)\,,((1,0),K)\}\,. \end{aligned}$$

The centers of mass of those two sets coincide and they are in optimal position with respect to rotations. The diameters of both sets are 2, so Algorithm 3.7 using the diameter leaves those two sets unchanged and computes

$$\mathrm{EMD}_2{}^{\mathrm{apx}}(A,B) \;=\; \frac{1}{2K}\,2\,(K-1) \;=\; \frac{K-1}{K}$$

as approximate EMD_2 under similarities. The optimal similarity scales B to 0 for $K > 2$ and $\mathrm{EMD}_2{}^{\mathrm{opt}}(A,B) = 1/K$. Thus we get that $\mathrm{EMD}_2{}^{\mathrm{apx}}(A,B)/\mathrm{EMD}_2{}^{\mathrm{opt}}(A,B) = K-1$, which tends to ∞ as K tends to ∞. This proves that there is no constant-factor approximation.

Normalized Second Moment Does Not Work

Similar as in the previous section we show that scaling by the quotient of the normalized second moments does not work either. Again we concentrate on the Euclidean case. We give the well-known definition of the normalized second moment of a weighted point set with respect to an arbitrary point $p^* \in \mathbb{R}^d$.

Definition 3.7 (Normalized Second Moment)**.** Let $A = \{(p_i,\alpha_i)_{i=1,\dots,n}\} \in \mathbb{W}^{d,G}$ be a weighted point set for some $G \in \mathbb{R}_{>0}$ and let $p^* \in \mathbb{R}^d$ be an arbitrary point. We call

$$m_{p^*}^{(2)}(A) \;=\; \frac{1}{W^A}\sum_{i=1}^{n} \alpha_i \,\|p_i - p^*\|^2$$

the normalized second moment of A with respect to p^*.

We show that normalizing by the second moment does not lead to a constant-factor approximation. We show that this does not work for the center of mass as an EMD-reference point and therefore cannot work in general.

Consider the following two sets, where $K \geq 1$ is any positive constant:

$$\begin{aligned} A &:= \left\{ \left((-\sqrt{K}, 0), 1 \right), \left((\sqrt{K}, 0), 1 \right), ((0,0), 2K-2) \right\}, \\ B &:= \{ ((-1,0), K), ((1,0), K) \}. \end{aligned}$$

The centers of mass of those two sets coincide and they are in optimal position with respect to rotations. The normalized second moments with respect to the origin can be computed as follows:

$$\begin{aligned} m_O^{(2)}(A) &= \frac{1}{2K} \left(\sqrt{K}^2 + \sqrt{K}^2 \right) = 1, \\ \text{and} \quad m_O^{(2)}(B) &= \frac{1}{2K} (K + K) = 1. \end{aligned}$$

Thus, Algorithm 3.7 using the quotient of the normalized second moments leaves the two weighted point sets unchanged and computes

$$\mathrm{EMD_2}^{\mathrm{apx}}(A, B) = \frac{1}{2K} \left(2\,(\sqrt{K} - 1) + 2\,(K - 1) \right),$$

which tends to 1 as K tends to ∞. The optimal similarity scales B to 0 for $K > 2$ and in this case we have

$$\mathrm{EMD_2}^{\mathrm{opt}}(A, B) = \frac{2\sqrt{K}}{2K},$$

which tends to 0 as K tends to ∞. Thus we get that $\mathrm{EMD_2}^{\mathrm{apx}}(A, B)/\mathrm{EMD_2}^{\mathrm{opt}}(A, B)$ tends to ∞ as K tends to ∞. This proves that there is no constant-factor approximation.

Chapter 4

Small Manhattan Networks

A Manhattan network on a set S of n points in the plane is a (not necessarily planar) rectilinear network G with the property that for every pair of points in S, the network G contains a path between them, whose length equals the Manhattan distance between the points. A Manhattan network on S can be thought of as a graph $G = (V, E)$, where the vertex set V corresponds to the points of S and a set of Steiner points S'. The edges in E correspond to horizontal and vertical line segments connecting points in $V = S \cup S'$. A Manhattan network can also be interpreted as a 1-spanner (for the L_1-metric) for the points in S. Note that in contrast to the other Chapters, a Steiner point here describes an additional vertex instead of the curvature centroid. We give the definition for arbitrary dimension:

Definition 4.1 (Rectlinear Network)**.** A rectilinear network is a graph $G = (V, E)$ with vertex set $V \subset \mathbb{R}^d$ and edge set E, such that E consists of axis-parallel line segments connecting the points in V.

Definition 4.2 (Manhattan Network)**.** Let $S = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$ be a set of points. A Manhattan network on S is a rectilinear network $G = (V, E)$, where V consists of the points of S and a set of Steiner points S', such that for every pair of points p_i, p_j in S there is a Manhattan path, i.e., a path of length $\|p_i - q_j\|_1$, between them.

The problem to compute a minimum length Manhattan network is a well-researched area, see Gudmundsson, Levcopoulos and Narasimhan [32], Benkert, Wolff, Widmann and Shirabe [13], and Chepoi, Nouioua and Vaxes [16]. Even though this problem has received considerable attention, the variant of minimizing the number of vertices and edges of the graph has not been considered.

In the planar case, we show that there is a Manhattan network on S with $O(n \log n)$ vertices and edges which can be constructed in $O(n \log n)$ time. The network constructed is not planar. We show that if we force the network to be planar, there are point sets where every Manhattan network needs $\Omega(n^2)$ vertices and edges. We further show that our construction is optimal in the sense that there are point sets in the plane where every Manhattan network needs $\Omega(n \log n)$ vertices and edges. At the expense of a slightly higher time and space complexity of $O(n \log^{d-1} n)$, we are able to extend our approach to any dimension $d \geq 3$. This allows us to compute the L_1-Earth Mover's Distance EMD_1 on weighted point sets in $\mathbb{R}^d$ in $O(n^2 \log^{2d-1} n)$ time, which improves the currently best known result of $O(n^4 \log n)$ using Orlin's algorithm, see Chapter 3.

Our approach can be used to speed up the reference point methods for the EMD introduced in the last chapter. While it can be directly applied in case of the EMD_1, it will increase the approximation ratio by a constant factor for other L_p-distances.

The main parts of this chapter have already been published by Gudmundsson, Klein, Knauer and Smid [33].

4.1 Small Manhattan Networks in the Plane

We formulate and prove the main result of this chapter, whose construction was suggested by Christian Knauer.

Theorem 4.1. *Let S be a set of n points in the plane. Then there is a Manhattan network on S with $O(n \log n)$ vertices and edges. It can be computed in $O(n \log n)$ time.*

Proof. Let $S = \{p_1, \ldots, p_n\}$ be a set of n points in the plane and assume that the points are sorted with respect to their y-coordinates. The sorting can be done in $O(n \log n)$ time. In the following, L always denotes a list of points which is sorted by y-coordinate. The i-th point in L is denoted by $L[i]$. We run the following algorithm on S, see Figure 4.1 for an illustration.

Algorithm 4.1 (ConstructNetwork(L))**.**

1. Find median p^* with respect to x-coordinate.
2. Set $L_1 := \emptyset$, $\quad L_2 := \emptyset$.
3. For $i = 1, \ldots, |L|$ do
   (a) Construct vertex $v[i] := (p^*_x, L[i]_y)$.
   (b) Construct edge $e_h[i] := (L[i], v[i])$.
   (c) If $i \geq 2$: Construct edge $e_v[i] := (v[i-1], v[i])$.
   (d) If $L[i]_x < p^*_x$: add $L[i]$ at the end of L_1.
   (e) If $L[i]_x > p^*_x$: add $L[i]$ at the end of L_2.
4. If $|L_1| > 1$: ConstructNetwork(L_1).
5. If $|L_2| > 1$: ConstructNetwork(L_2).

We have to prove that the algorithm constructs a Manhattan network. Let $p, q \in S$ be two arbitrary points. Let p^* be the first point chosen as a median in Step 1 with $p_x \leq p^*_x \leq q_x$. Clearly, p and q are both contained in L. W.l.o.g., let $p = L[i] =: p_i$ and $q = L[j] =: p_j$ with $i < j$. In Step 3, p_i is considered before p_j. Therefore, by construction there are vertices $v[i], v[j]$, edges $(v[i], p_i)$, $(v[j], q_j)$ and sequence of Steiner points $v[i], \ldots, v[j]$. This sequence is y-monotone since we have considered the original points in y-increasing order. Thus, the whole sequence $p_i, v[i], \ldots, v[j], p_j$ is an x- and y-monotone path consisting of two horizontal edges connected by a path of vertical edges, and therefore this path is a Manhattan path. This proves that the resulting graph is a Manhattan network on S.

The median in a list of $k := |L|$ numbers can be computed in $O(k)$ time using a result by Blum, Floyd, Pratt, Rivest and Tarjan [14]. Steps (a) to (d) can be done in constant time. Therefore, the runtime of Algorithm 4.1 without the two recursive calls is $O(k)$. The insertion in the lists L_1, L_2 is done in sorted order with respect to the y-coordinate. No re-sorting is needed after the initial sorting step.

The number of points in the lists L_1 and L_2 is at most half the number of points in L. Thus, the overall runtime can be described by the recursion $T(n) = O(n) + 2 \cdot T(n/2)$, which

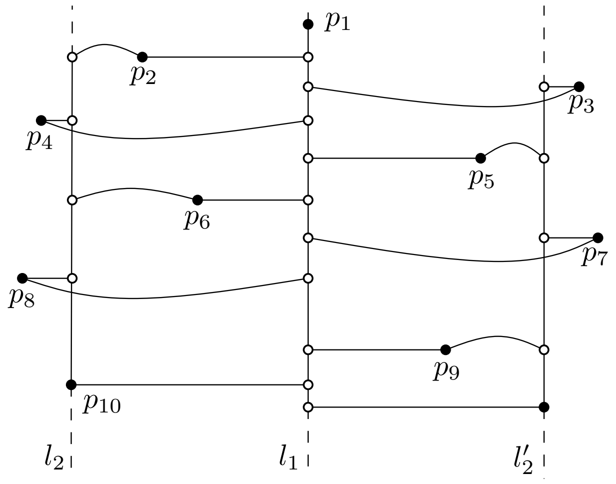


Figure 4.1: Construction of the Network.

resolves to $T(n) = O(n \log n)$. The number of Steiner points and edges in the construction obeys the same recursion, since in every recursive call of Algorithm 4.1, $O(k)$ vertices and edges are added. □

In practice, paths with a small number of vertices are often desirable. We show how to construct a network such that for every pair of points there is a shortest path with a small number of vertices. Let $\alpha(n)$ denote the inverse of Ackermann's function, see Yao [53].

Theorem 4.2. *Let S be a set of n points in the plane. Then there is a Manhattan network on S with $O(n \log n)$ vertices and edges, where the number of edges on a shortest Manhattan path between any pair of points is bounded by $O(\alpha(n))$. The network can be computed in $O(n \log n)$ time.*

Proof. The Manhattan path between two input points p, q constructed by Algorithm 4.1 always has the form $p, v_1, \ldots, v_k, p_j$, where $v_1, \ldots, v_k$ is a y-monotone sequence of Steiner points lying on a vertical line. Let k^* be the number of Steiner points lying on such a vertical line, generated in one recursion step in Algorithm 4.1. Using a result of Yao [53], we can compute $O(k^*)$ edges in $O(k^*)$ time, each connecting two Steiner points, such that for any pair of Steiner points the number of edges on a y-monotone shortest path is $O(\alpha(k^*))$. That is, there is a Manhattan path between p and q with length $O(\alpha(k^*)) + 2 = O(\alpha(n))$. Since we can compute these $O(k^*)$ edges in every recursive call in $O(k^*)$ time, the asymptotic runtime, the number of edges and the number of Steiner points does not change. □

At the expense of a slightly higher runtime we can reduce the length of a shortest Manhattan path to a constant. The proof is analogous to the proof of Theorem 4.2, again using constructions by Yao [53].

Theorem 4.3. *Let S be a set of n points in the plane. Then there is a Manhattan network on S with $O(n \log^2 n)$, $O(n \log n \log \log n)$ and $O(n \log n \log^* n)$ vertices and edges, where the*

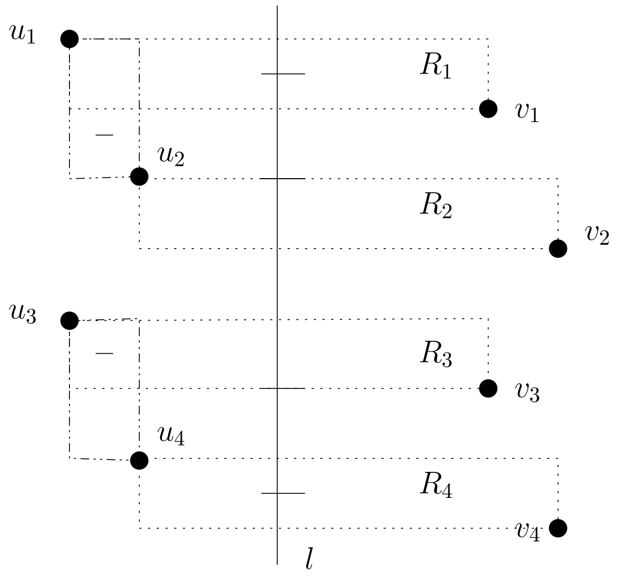


Figure 4.2: Lower Bound

number of edges on a shortest Manhattan path between any pair of points is bounded by 6, 7 *and* 8*, respectively. The networks can be computed in the same runtime.*

The upper bound given in Theorem 4.1 is tight. We show this by using 2-dimensional Horton sets which were introduced by Horton [34].

Theorem 4.4. *There are sets of n points in the plane, where every Manhattan network needs $\Omega(n \log n)$ vertices and edges.*

Proof. We construct a point set P in general position, such that any Manhattan network on P consists of $\Omega(n \log n)$ vertices and edges. We assume that n is a power of two. Let ℓ be a vertical line separating P into two point sets $U := \{u_1, \ldots, u_{n/2}\}$ and $V := \{v_1, \ldots, v_{n/2}\}$, such that the points $u_1, v_1, u_2, v_2, \ldots, u_{n/2}, v_{n/2}$ are sorted by y-coordinates, from top to bottom, see Figure 4.2.

For $1 \leq i \leq n/2$, let R_i be the axis-parallel rectangle with top-left corner u_i and bottom-right corner v_i. Any Manhattan network on P must contain a path between u_i and v_i that crosses ℓ and is completely contained in R_i. Since the rectangles R_i are pairwise disjoint, it follows that any Manhattan network on P contains at least $n/2$ edges that cross ℓ. Observe that this remains true if we move the points of U and V horizontally, as long as U stays to the left of ℓ and V stays to the right of ℓ. Thus, we can move the points of U, such that they can be split into two subsets U_1 and U_2 that are separated by a vertical line ℓ' such that the sorted y-order alternates between a point in U_1 and a point in U_2. Any Manhattan network on P must contain at least $n/4$ edges that cross ℓ' and that are distinct from the above $n/2$ edges. Similarly, we can move the points of V, and split them into two subsets V_1 and V_2 that are separated by a vertical line ℓ'' in such a way that any Manhattan network on P must contain at least $n/4$ edges that cross ℓ'' and are distinct from the above $n/2 + n/4$ edges. We continue this moving in a recursive way. Below we prove that all these edges are distinct. It follows that the number $T(n)$ of vertices and edges in any Manhattan network on the final set P satisfies $T(n) \geq n/2 + 2 \cdot T(n/2)$, which proves that $T(n) = \Omega(n \log n)$. □

A rigorous proof that the edges constructed in the last proof are distinct was given by Smid [33]:

Proof. For every integer n which is a power of two, we define the following $n \times \log n$ matrix M_n: Each entry of M_n is one of the symbols L and R. For $1 \leq k \leq \log n$, the k-th column of M_n, when considered as a string over the alphabet $\{L, R\}$, is equal to

$$\left(L^{2^{k-1}} R^{2^{k-1}}\right)^{n/2^k}.$$

Observe that $M_2 = \binom{L}{R}$ and, for $n \geq 4$, M_n consists of two copies of $M_{n/2}$ stacked on top of each other and one last column which is equal to the transpose of $(L^{n/2}R^{n/2})$.

Let T_n be a perfectly balanced binary tree with n leaves. We store the values $1, 2, \ldots, n$ at these leaves (from left to right).

Define the point set $P_n = \{p_1, p_2, \ldots, p_n\}$ in the following way: For $1 \leq i \leq n$, the y-coordinate of p_i is equal to $n - i$. To define the x-coordinate of p_i, we traverse the tree T_n (starting at the root) and follow the path as described by the i-th row of the matrix M_n (where, of course, L stands for "go left" and R stands for "go right"). The x-coordinate of p_i is equal to the value stored at the leaf in which this path ends.

Observe that the set P_2 consists of the two points $(1, 1)$ and $(2, 0)$. For $n \geq 4$, the point set P_n consists of two "copies" U and V of $P_{n/2}$, such that

1. U has the same geometric structure as $P_{n/2}$,
2. V has the same geometric structure as $P_{n/2}$,
3. U is completely to the left of V, and
4. when visiting the points of P_n from top to bottom, we alternate between a point in U and a point of V.

Let $T(n)$ denote the minimum number of edges in any Manhattan network on P_n. Obviously, $T(2) \geq 2$. Let $n \geq 4$, and consider an arbitrary Manhattan network G_n on P_n. Consider the two point sets U and V as described above. Let ℓ be the vertical line that separates U and V. The edge set of the network G_n consists of

1. an edge set E_L containing all edges that are completely to the left of ℓ,
2. an edge set E_R containing all edges that are completely to the right of ℓ, and
3. an edge set E_{LR} containing all edges that cross ℓ.

The edge set E_L defines a graph on the set U, which is a Manhattan network on U. Since the set U has the same geometric structure as $P_{n/2}$, the size of E_L is at least $T(n/2)$. Similarly, the size of E_R is at least $T(n/2)$. To prove a lower bound on the size of E_{LR}, we write $U = \{u_1, \ldots, u_{n/2}\}$ and $V = \{v_1, \ldots, v_{n/2}\}$, such that the points $u_1, v_1, u_2, v_2, \ldots, u_{n/2}, v_{n/2}$ are sorted by y-coordinates, from top to bottom. For $1 \leq i \leq n/2$, let R_i be the axis-parallel rectangle with top-left corner u_i and bottom-right corner v_i. The network G_n must contain a

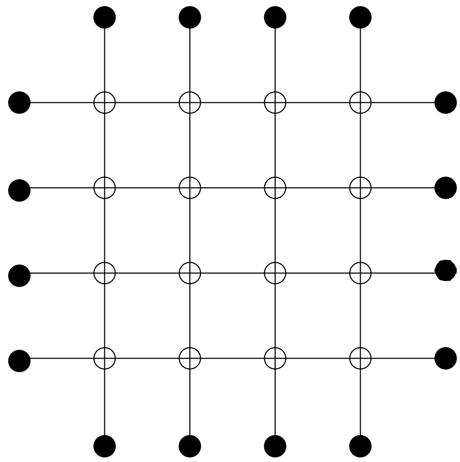


Figure 4.3: Planar Construction.

path between u_i and v_i that crosses ℓ and that is completely contained in R_i. In particular, this path contains an edge which is in E_{LR}. Since the rectangles R_i are pairwise disjoint, it follows that E_{LR} contains at least $n/2$ edges. Thus, we have shown that the network G_n contains at least $n/2 + 2 \cdot T(n/2)$ edges. Since G_n was an arbitrary network on P_n, it follows that $T(n) \geq n/2 + 2 \cdot T(n/2)$, which proves that $T(n) = \Omega(n \log n)$. □

If the network is required to be planar the lower bound can be improved:

Theorem 4.5. *There are sets of n points in the plane, where every planar Manhattan network needs $\Omega(n^2)$ vertices and edges.*

Proof. Let the set P of points in $\mathbb{R}^2$ be defined as follows, see the black points in Figure 4.3:

$$P := \bigcup_{i=1}^{n-1} \left(\left\{ \left(\frac{i}{n}, 0 \right) \right\} \cup \left\{ \left(\frac{i}{n}, 1 \right) \right\} \cup \left\{ \left(0, \frac{i}{n} \right) \right\} \cup \left\{ \left(1, \frac{i}{n} \right) \right\} \right).$$

Let G be a Manhattan network for this point set. There must be a Manhattan path between every pair of points $(\frac{i}{n}, 0), (\frac{i}{n}, 1)$ and $(0, \frac{i}{n}), (1, \frac{i}{n})$. These paths have to be straight lines, since in the first case the x-coordinate and in the second case the y-coordinate is the same. This forces the $\Omega(n^2)$ intersection points of the straight lines to be Steiner points. □

A point set in general position giving the same lower bound can be constructed easily by perturbing the points slightly.

4.2 Higher Dimensions

In dimensions $d \geq 3$ we can use a similar divide-and-conquer approach as in the plane.

Theorem 4.6. *Let S be a set of n points in $\mathbb{R}^d$. Then there is a Manhattan network on S with $O(n \log^{d-1} n)$ vertices and edges. It can be computed in $O(n \log^{d-1} n)$ time.*

Proof. Let $S = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$ be a set of n d-dimensional points and assume that the points are sorted with respect to their first coordinates. The sorting can be done in $O(n \log n)$ time. In the following, L always denotes a list of points which is sorted by their first coordinate. The i-th point in L is denoted by $L[i]$. Run the following algorithm on S.

Algorithm 4.2 (ConstructNetwork(L, d)).

1. Find median p^* with respect to the d-th coordinate.
2. Set $L_1 := \emptyset, \quad L_2 := \emptyset, \quad \mathcal{P} := \emptyset$.
3. For $i = 1, \ldots, |L|$ do
   (a) Project $L[i]$ on the hyperplane containing p^* and orthogonal to the d-th coordinate.
   (b) Let Proj($L[i]$) be this point.
   (c) Construct edge ($L[i]$, Proj($L[i]$)).
   (d) Add Proj($L[i]$) at the end of $\mathcal{P}$.
   (e) If $d = 2$ and $i \geq 2$: Construct edge (Proj($L[i-1]$), Proj($L[i]$)).
   (f) If $L[i]_d < p^*_d$: add $L[i]$ at the end of L_1.
   (g) If $L[i]_d > p^*_d$: add $L[i]$ at the end of L_2.
4. If $d > 2$: ConstructNetwork($\mathcal{P}, d-1$).
   (Compute the Manhattan network on this hyperplane.)
5. If $|L_1| > 1$: ConstructNetwork(L_1, d).
6. If $|L_2| > 1$: ConstructNetwork(L_2, d).

Except for the recursive calls in the algorithm, any call can be done in $O(|L|)$ time. There are three recursive calls, one call of the routine for the same number of points in one dimension less and two calls for the number of points halved in the same dimension. Analogous to the earlier proof, the runtime of this can be expressed as

$$\begin{aligned} T(n,d) &= O(n) + T(n, d-1) + 2 \cdot T(n/2, d) \\ &= O(n \log^{d-1} n). \end{aligned}$$

The bound on the number of points and edges follows analogously. Note also that the sorting of the points with respect their first coordinates is maintained and no re-sorting is necessary. ☐

4.3 Earth Mover's Distance

We now show how we can use Algorithm 4.2 to compute the L_1-Earth Mover's Distance (EMD$_1$) on weighted point sets in $O(n^2 \log^{2d-1} n)$ time. This improves the previously best known time of $O(n^4 \log n)$. See Chapter 3 for a definition and more details on the EMD.

Theorem 4.7. *The* EMD$_1$ *can be computed in* $O(n^2 \log^{2d-1} n)$ *time.*

Proof. Let A, B be weighted point sets. Using Theorem 4.6 we can construct a 1-spanner of the complete bipartite graph between the points of A and B for the L_1-metric in $O(n \log^{d-1} n)$ time. The number of points and edges in the resulting network is bounded by $O(n \log^{d-1} n)$. Now we proceed as in Cabello, Giannopoulos, Knauer and Rote [15]. By the standard method of doubling each edge and orienting the two copies in different directions we obtain a flow network, where between any pair of points there is a directed path of minimum L_1-length. Now we can use the minimum cost flow algorithm by Orlin [41] on the 1-spanner. Given a network $G = (V, E)$, Orlin's algorithm solves the minimum cost flow problem in $O((|E| \log |V|)(|E| + \log |V|))$. Since the number of points and edges in our spanner is bounded by $|E| = |V| = O(n \log^{d-1} n)$, the overall runtime is bounded by $O(n^2 \log^{2d-1} n)$. □

Theorem 4.7 immediately leads to a $\sqrt{2}$-approximation with the same runtime for the important case when the EMD is based on the Euclidean distance. This algorithm is conceptually easier than the slightly faster $(1 + \varepsilon)$-approximation given by Cabello, Giannopoulos, Knauer and Rote [15]. Their method has a runtime of $O((n^2/\varepsilon^2) \log^2 n)$ in arbitrary dimension.

Chapter 5

Monge-Kantorovich Distance

In Chapter 3 we have seen how to apply our reference point framework for matching weighted point sets with respect to the Earth Mover's Distance (EMD). In this chapter we discuss whether it is possible to apply the framework to the more general case of Borel measures defined on $\mathbb{R}^d$. The corresponding distance measure is called the Monge-Kantorovich Distance (MKD). For a comprehensive introduction into this topic see Rachev and Rüschendorf [43], and Villani [50]. We first show that the center of mass is an MKD-reference point for Borel measures with equal total weight with respect to affine transformations. This was already indicated by Cohen [19]. We further show that we can use reference points in the same way as we did in the discrete case to construct approximation algorithms for matching Borel measures under translations. Unfortunately we cannot directly generalize the constant-factor approximation algorithms for rigid motions and similarities. In these cases we prove approximation algorithms for bounding the absolute error.

Although we give a short introduction into Borel measures, we assume basic knowledge of integration theory. A short introduction can be found in the book by Villani [50].

5.1 Definition

We introduce measures on the Borel σ-algebra $\mathcal{B}$ in $\mathbb{R}^d$.The Borel σ-algebra $\mathcal{B}$ of Borel sets is defined as the σ-algebra generated by the open subsets in $\mathbb{R}^d$. That is, $\mathcal{B}$ fulfills:

1. $\emptyset \in \mathcal{B}$
2. $A \subset \mathbb{R}^d$ open set $\Rightarrow A \in \mathcal{B}$
3. $A \in \mathcal{B} \Rightarrow \mathbb{R}^d \setminus A \in \mathcal{B}$
4. $(A_i)_{i\in\mathbb{N}} \in \mathcal{B} \Rightarrow \bigcup_{i=1}^{\infty} A_i \in \mathcal{B}$

A Borel measure is a non-negative and countably additive function defined on the Borel sets:

Definition 5.1 (Borel Measure)**.** A function $\varphi\colon \mathcal{B} \to \mathbb{R}_{\geq 0}$ is called a Borel measure, if $\varphi(\emptyset) = 0$, and for pairwise disjoint sets of elements $(A_i)_{i\in\mathbb{N}} \in \mathcal{B}$ it holds that

$$\varphi\left(\bigcup_{i\in\mathbb{N}} A_i\right) = \sum_{i\in\mathbb{N}} \varphi(A_i).$$

As in the case of the EMD, our matching algorithms for two given Borel measures φ and ψ only work when the total weights $\varphi(\mathbb{R}^d)$ and $\psi(\mathbb{R}^d)$ are equal. For simplicity, we concentrate on Borel measures with total weight 1, that is, Borel probability measures:

Definition 5.2 (Borel Probability Measure)**.** A function $\varphi\colon \mathcal{B} \to \mathbb{R}_{\geq 0}$ is called a Borel probability measure, if φ is a Borel measure and $\varphi(\mathbb{R}^d) = 1$.

The extension of the following algorithms to arbitrary Borel measures with finite positive total weight is straightforward by normalizing by the coinciding total weight of the measures.

Let $A, B \in \mathcal{B}$ be Borel sets. We can imagine $\varphi(A)$ as the amount of mass which is located in A, and $\psi(B)$ as the amount of mass which can be put into B. To define the MKD between two given measures we need to define a transportation plan between them. A transportation plan is a Borel probability measure π defined on the Borel sets of $\mathbb{R}^d \times \mathbb{R}^d$, where $\pi(A \times B)$ denotes the amount of mass which is moved away from A and put into B. To get a reasonable transportation, $\pi(A \times \mathbb{R}^d)$ describes the total amount of mass which is moved away from A. Since the total weights of φ and ψ are the same, $\pi(A \times \mathbb{R}^d)$ equals the amount of mass $\varphi(A)$ which is located in A. Similarly, $\pi(\mathbb{R}^d \times B)$ equals the amount of mass which is put into B and therefore equals $\psi(B)$. This leads to the following definition:

Definition 5.3 (Transportation Plan)**.** Let φ, ψ be Borel probability measures. A transportation plan from φ to ψ is a Borel probability measure π defined on the Borel sets of $\mathbb{R}^d \times \mathbb{R}^d$, such that for all Borel sets $A, B \in \mathcal{B}$ we have:

1. $\pi(A \times \mathbb{R}^d) = \varphi(A)$
2. $\pi(\mathbb{R}^d \times B) = \psi(B)$

Let $\Pi(\varphi, \psi)$ denote the set of all transportation plans from φ to ψ. Note that $\Pi(\varphi, \psi) \neq \emptyset$ since the tensor product of φ and ψ is in $\Pi(\varphi, \psi)$. This plan corresponds to the transportation distributing every single piece of mass over the entire hole, proportionally to the depth.

We define the Monge-Kantorovich Distance between two measures in the following way:

Definition 5.4 (Monge-Kantorovich Distance)**.** Let φ, ψ be Borel probability measures. We call

$$\mathrm{MKD}(\varphi, \psi) := \inf_{\pi\in\Pi(\varphi,\psi)} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|v - w\| \, d\pi(v, w)$$

the Monge-Kantorovich Distance between φ and ψ.

As in the case of the EMD we do not specify the norm on the underlying space $\mathbb{R}^d$ in the above definition. In fact, some of our algorithms work for any norm, although sometimes adjustments on the approximation ratios are necessary. In some cases the results only hold when the underlying space $\mathbb{R}^d$ is equipped with an L_p-distance, where $1 \leq p \leq \infty$. In these cases we also use MKD_p to denote the Monge-Kantorovich distance.

In the following sections we minimize the MKD under translations, rigid motions and similarities. Let S be a similarity and φ be any probability measure. Then we define the transformed probability measure $S(\varphi)$ by the function φ', where

$$\varphi' \colon \mathcal{B} \to \mathbb{R}_{\geq 0}, \quad A \mapsto \varphi(S(A)),$$

i.e., $\varphi'(A)$ equals the measure of the transformed Borel set $S(A)$.

5.2 The Center of Mass as an MKD-Reference Point

As in the discrete case of the EMD we prove that the center of mass of a Borel probability measure is an MKD-reference point with respect to affine transformations. We first define the continuous version of the center of mass:

Definition 5.5 (Center of Mass)**.** Let $\varphi \colon \mathcal{B} \to \mathbb{R}_{\geq 0}$ be a Borel probability measure. The center of mass of φ is defined as

$$C(\varphi) \ := \ \int_{\mathbb{R}^d} v \, d\varphi(v).$$

The main part of the following theorem is the Lipschitz continuity of the center of mass with respect to the MKD. Cohen [19] proves this in the discrete case and indicates the extendability to the continuous case.

Theorem 5.1. *The center of mass is an* MKD*-reference point for Borel probability measures with respect to affine transformations. Its Lipschitz constant is* 1*. This holds for arbitrary dimension d and any norm on the underlying space $\mathbb{R}^d$.*

Proof. Let φ, ψ be Borel probability measures. The equivariance of the center of mass of a Borel probability measure under affine transformations is well-known. To prove the Lipschitz continuity we have to show that

$$\|C(\varphi) - C(\psi)\| \ \leq \ \mathrm{MKD}(\varphi, \psi).$$

By definition we have

$$\|C(\varphi) - C(\psi)\| \ = \ \left\| \int_{v \in \mathbb{R}^d} v \, d\varphi(v) - \int_{w \in \mathbb{R}^d} w \, d\psi(w) \right\|.$$

Let π be a transportation plan determining $\mathrm{MKD}(\varphi, \psi)$. Informally, $d\varphi(v)$ equals the weight located at the point $v \in \mathbb{R}^d$ and therefore equals the amount of mass moved by π to all points $w \in \mathbb{R}^d$, which is described by $\int_{w \in \mathbb{R}^d} d\pi(v, w)$. Analogously, we have that $d\psi(w)$ equals $\int_{v \in \mathbb{R}^d} d\pi(v, w)$. Therefore,

$$\begin{aligned}
\|C(\varphi) - C(\psi)\| \ &= \ \left\| \int_{v \in \mathbb{R}^d} v \int_{w \in \mathbb{R}^d} d\pi(v, w) - \int_{w \in \mathbb{R}^d} w \int_{v \in \mathbb{R}^d} d\pi(v, w) \right\| \\
&= \ \left\| \int_{v \in \mathbb{R}^d} \int_{w \in \mathbb{R}^d} v \, d\pi(v, w) - \int_{v \in \mathbb{R}^d} \int_{w \in \mathbb{R}^d} w \, d\pi(v, w) \right\| \\
&= \ \left\| \int_{v \in \mathbb{R}^d} \int_{w \in \mathbb{R}^d} (v - w) \, d\pi(v, w) \right\| \\
&\leq \ \int_{v \in \mathbb{R}^d} \int_{w \in \mathbb{R}^d} \|v - w\| \, d\pi(v, w) \\
&= \ \mathrm{MKD}(\varphi, \psi).
\end{aligned}$$

This proves that the Lipschitz constant of the center of mass as an MKD-reference point is at most 1. The lower bound follows directly by the fact that the Lipschitz constant of any reference point is at least 1, see Theorem 1.4. □

We prove the following basic but fundamental result which we use to apply the abstract approximation algorithm for translations, see Section 1.5.

Theorem 5.2. *Let φ be a Borel probability measure and let $\tau \in \mathbb{R}^d$ be a translation. Then*

$$\mathrm{MKD}\,(\varphi, \tau(\varphi)) \;=\; \|\tau\|.$$

Proof.

$$\begin{aligned}
&\mathrm{MKD}(\varphi, \tau(\varphi)) \\
&= \min_{\pi \in \Pi(\varphi, \tau(\varphi))} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|v - w\| \, d\pi(v, w) \\
&= \min_{\pi \in \Pi(\varphi, \varphi)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|v - (w + \tau)\| \, d\pi(v, w) \\
&\leq \int_{\mathbb{R}^d} \|v - (v + \tau)\| \, d\varphi(v), \quad \text{by choosing } \pi(v, v) := \varphi(v) \text{ as transportation plan} \\
&= \|\tau\| \int_{\mathbb{R}^d} d\varphi(v) \\
&= \|\tau\|.
\end{aligned}$$

By Lipschitz continuity of the center of mass we have $\mathrm{MKD}(\varphi, \tau(\varphi)) \geq \|C(\varphi) - C(\tau(\varphi))\| = \|\tau\|$. This proves the lemma. □

5.3 Approximation Using MKD-Reference Points

In this section we give approximation algorithms for the MKD under translations, rigid motions and similarities. The section is organized as follows: In each part we consider a class of transformations, construct an approximation algorithm for matching under these transformations for general MKD-reference points, and finally use the center of mass to obtain a concrete algorithm. The following results do not hold for measures with unequal total weight.

In the case of rotations, rigid motions and similarities we concentrate on the MKD based on the Euclidean distance on the underlying space $\mathbb{R}^d$. The extension to arbitrary L_p-distances is similar to the case of the EMD and does not lead to any new insights.

In this section let $\varphi, \psi \colon \mathcal{B} \to \mathbb{R}_{\geq 0}$ be Borel probability measures. Let r be an MKD-reference point with respect to the considered class of transformations and with Lipschitz constant c. Let T^{ref} be the time to compute the MKD-reference point and T^{MKD} the time to compute the MKD.

5.3.1 Translations

In this section we apply the abstract approximation algorithms for translations introduced in Sections 1.5.1 and 1.5.2.

2-Approximation Algorithm

The following algorithm finds an approximation for the MKD under translations. It determines the reference points and computes the MKD once. In general there is no algorithm to compute the MKD, even for fixed probability measures. However, there may be applications where an achievement of similarity by a translation of one of the sets is desired. The following algorithm provides a way to find such a translation without computing the MKD.

Algorithm 5.1.

1. Compute $r(\varphi)$ and $r(\psi)$ and translate ψ by $r(\varphi) - r(\psi)$. Let ψ' be the image of ψ.
2. Output ψ' together with the distance $\mathrm{MKD}(\varphi, \psi')$.

The following theorem is a direct consequence of Theorems 1.5 and 5.2.

Theorem 5.3. *Algorithm 5.1 finds an approximately optimal matching for translations with approximation factor $c+1$ in time $O(T^{\mathrm{ref}} + T^{\mathrm{MKD}})$. This holds for arbitrary dimension d and any norm on the underlying space $\mathbb{R}^d$.*

The Lipschitz constant of the center of mass is 1 and the time to compute this point is dominated by the time to compute the MKD. Therefore we can state the following corollary:

Corollary 5.1. *Algorithm 5.1 using the center of mass as an* MKD*-reference point induces an approximation algorithm with approximation factor* 2*. The time to compute the approximation is $O(T^{\mathrm{MKD}})$.*

Lower Bound for the Approximation Factor of Algorithm 5.1

The approximation factor given in Corollary 5.1 is tight in the Euclidean case. Recall that we write MKD_2 for the Euclidean Monge-Kantorovich distance.

Theorem 5.4. *There are Borel probability measures where the upper bound on the approximation factor of Algorithm 5.1 using the center of mass as an* MKD_2*-reference point is assumed in the limit.*

Proof. The result follows by considering small environments around the weighted points which prove the lower bound for the corresponding algorithm for the EMD_2, see Section 3.4.1. The probability measure can be any function distributing $1/K$ of the weight in one of the environments and $(K-1)/K$ in the other. □

Fully Polynomial-Time Approximation Scheme for Translations

We apply the abstract $(1+\varepsilon)$-approximation algorithm presented in Section 1.5.2 to the MKD_p. Thus, using Theorems 1.6 and 5.2 we see the following result.

Theorem 5.5. *Let φ, ψ be Borel probability measures. Let $1 \leq p \leq \infty$. Let r be an* MKD_p*-reference point with respect to translations and with Lipschitz constant c. Then there exists an algorithm that for any $0 < \varepsilon < 1$ finds a translation τ^{ε}, such that*

$$\mathrm{MKD}_p(\varphi, \tau^{\varepsilon}(\psi)) \;\leq\; (1+\varepsilon) \cdot \mathrm{MKD}_p(\varphi, \tau^{\mathrm{opt}}(\psi)).$$

Its runtime is $O(\varepsilon^{-d} \cdot T^{\mathrm{MKD}_p})$.

5.3.2 Rotations

In this section we consider the problem to find an approximation for the MKD under rotations around a fixed point $p^* \in \mathbb{R}^d$. Unfortunately, we are only able to bound the absolute error. That is, the result does not lead to a constant-factor approximation. We find the approximate rotation by computing the minimum MKD on a sufficiently large and dense grid in rotation space. Let $K > 0$ be the desired bound on the absolute error. Then the number of grid points we use is $O((K^{-1} \cdot m_{p^*}(\psi))^{d-1})$, where

$$m_{p^*}(\psi) \ := \ \int_{v \in \mathbb{R}^d} \|v - p^*\|_2 \, d\psi(v)$$

denotes the first moment of ψ with respect to p^*. See Definition 3.6 for the discrete analogue.

We first compute an approximation for planar measures with respect to the Euclidean MKD_2 and address the general case later. Let R_γ denote the rotation by angle $\gamma \in [-\pi, \pi]$ around p^*.

Algorithm 5.2.

1. Let $\alpha := 2 \cdot \arcsin\left(K/(2 \cdot m_{p^*}(\psi)\right)$.
2. $\beta^* := \arg\min \{ \, \mathrm{MKD}(\varphi, R_\beta(\psi)) \ : \ \beta = k\alpha \text{ with } k \in \mathbb{Z}, \text{ and } \beta \in [-\pi, \pi] \, \}$
3. Output $R_{\beta^*}(\psi)$ together with the distance $\mathrm{MKD}(\varphi, R_{\beta^*}(\psi))$.

Theorem 5.6. *Let φ, ψ be Borel probability measures in the plane, and let $K > 0$. Algorithm 5.2 finds a rotation $R^{\mathrm{apx}} := R_{\beta^*}$, such that*

$$\mathrm{MKD}_2(\varphi, R^{\mathrm{apx}}(\psi)) \ \leq \ K + \min_{R \in \mathrm{Rot}(p^*)} \mathrm{MKD}_2(\varphi, R(\psi)).$$

The runtime is the time to compute the MKD_2 at $O(K^{-1} \cdot m_{p^*}(\psi))$ angles in rotation space.

Proof. W.l.o.g. let φ, ψ be in optimal position with respect to rotations of ψ around p^*. Let $\alpha = 2 \cdot \arcsin(K/(2 \cdot m_{p^*}(\psi))$ denote the angle between two grid angles in rotation space. We assume that $\alpha \leq \pi/2$. If necessary we can achieve this by adding at most 4 angles in rotation space. Let β' be the grid angle in rotation space closest to the optimum. W.l.o.g. let $\beta' \geq 0$.

$$\begin{aligned}
&\mathrm{MKD}_2(\varphi, R^{\mathrm{apx}}(\psi)) \\
&\leq \ \mathrm{MKD}_2\left(\varphi, R_{\beta'}(\psi)\right), \quad \text{since } \beta' \text{ is a grid point} \\
&\leq \ \mathrm{MKD}_2\left(\varphi, \psi\right) + \mathrm{MKD}_2\left(\psi, R_{\beta'}(\psi)\right) \\
&= \ \mathrm{MKD}_2\left(\varphi, \psi\right) + \min_{\pi \in \Pi(\psi, R_{\beta'}(\psi))} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|v - w\|_2 \, d\pi(v, w) \\
&\leq \ \mathrm{MKD}_2\left(\varphi, \psi\right) + \int_{\mathbb{R}^d} \left\|v - R_{\beta'}(v)\right\|_2 \, d\psi(v), \quad \text{by choosing a fixed transportation plan} \\
&= \ \mathrm{MKD}_2\left(\varphi, \psi\right) + \int_{\mathbb{R}^d} 2 \cdot \sin(\beta'/2) \, \left\|v - p^*\right\|_2 \, d\psi(v) \qquad (5.1) \\
&= \ \mathrm{MKD}_2\left(\varphi, \psi\right) + 2 \cdot \sin(\beta'/2) \cdot m_{p^*}(\psi) \\
&\leq \ \mathrm{MKD}_2\left(\varphi, \psi\right) + 2 \cdot \sin(\alpha/2) \cdot m_{p^*}(\psi), \quad \text{since } 0 \leq \beta'/2 \leq \alpha/2 \leq \pi/4 \\
&= \ \mathrm{MKD}_2(\varphi, \psi) + 2 \, \frac{K}{2 \cdot m_{p^*}(\psi)} \cdot m_{p^*}(\psi), \quad \text{by definition of } \alpha \\
&= \ \mathrm{MKD}_2(\varphi, \psi) + K.
\end{aligned}$$

The number of grid angles is

$$O\left(\frac{2\pi}{\alpha}\right) \;=\; O\left(\frac{2\pi}{2\cdot\arcsin(K/(2\cdot m_{p^*}(\psi))}\right) \;=\; O\left(\frac{m_{p^*}(\psi)}{K}\right).$$

□

The angle between the grid angles in the above theorem is given by $\alpha = 2\cdot\arcsin(K/(2\cdot m_{p^*}(\psi))$ and the runtime of the algorithm is $O(m_{p^*}(\psi)/K)$. Since

$$\min_{R\in\mathrm{Rot}(p^*)} \mathrm{MKD}_2(\varphi, R(\psi)) \;=\; \min_{R\in\mathrm{Rot}(p^*)} \mathrm{MKD}_2(R(\varphi), \psi),$$

we can compute the optimal rotation for the probability measure with the smaller first moment and thereby improve the runtime of the algorithm.

We generalize the last result to higher dimensions by using a sufficiently large grid on the boundary of the d-dimensional unit ball to determine the grid angles. The number of grid angles is exponential in the dimension.

Theorem 5.7. *Let φ, ψ be Borel probability measures in $\mathbb{R}^d$, and let $K > 0$. We can find a rotation R^{apx}, such that*

$$\mathrm{MKD}_2(\varphi, R^{\mathrm{apx}}(\psi)) \;\le\; K + \min_{R\in\mathrm{Rot}(p^*)} \mathrm{MKD}_2(\varphi, R(\psi)).$$

The runtime is the time to compute the MKD_2 *at $O((K^{-1}\cdot m_{p^*}(\psi))^{d-1})$ angles in rotation space.*

Proof. Let $\delta := K/m_{p^*}(\varphi)$. Let Y be a δ-net on the boundary of the d-dimensional unit ball $\mathbb{S}^{d-1}$. In general, a δ-net on $\mathbb{S}^{d-1}$ is a subset $Y \subset \mathbb{S}^{d-1}$, such that for any point $x \in \mathbb{S}^{d-1}$ there exists a point $y \in Y$ with $\|x-y\|_2 \le \delta$, see Gonzalez [29] and Clarkson [18]. A δ-net can be computed by a greedy-algorithm with a runtime linear in the size of the net. This δ-net consists of $\Theta(1/\delta^{d-1})$ points on $\mathbb{S}^{d-1}$.

Let R^{apx} be a rotation inducing the minimum $\mathrm{MKD}_2(\varphi, R_\beta(\psi))$, minimized over all angles β corresponding to points in Y. Let g be the grid angle closest to the optimal rotation R^{opt}. Let β' be the angle between g and R^{opt}. Then,

$$\sin(\beta'/2) \;=\; \|g - R^{\mathrm{opt}}\|_2/2 \;\le\; \delta/2 \;=\; K/(2\cdot m_{p^*}(\varphi)).$$

Substituting "=" by "≤" in equation (5.1) we can use an analogous calculation as in the proof of the planar case and the claim follows. □

The latter two results can be generalized to arbitrary L_p-distances. The proof is similar to the proof of Lemma 3.5 and does not lead to new insights.

5.3.3 Rigid Motions

The approach to construct an approximation algorithm for rigid motions is similar to the construction of the abstract algorithm in Section 1.5.3. Of course, small differences exist based on the absence of a constant-factor approximation for rotations in general. The algorithm combines the approximation algorithms for translations and rotations:

Algorithm 5.3.

1. Compute $r(\varphi)$ and $r(\psi)$ and translate ψ by $r(\varphi) - r(\psi)$. Let ψ' be the image of ψ.

2. Determine a rotation $R^{\text{apx}} \in \text{Rot}(r(\varphi))$, such that
$$\text{MKD}(\varphi, R^{\text{apx}}(\psi')) \;\leq\; K + \min_{R \in \text{Rot}(r(\varphi))} \text{MKD}(\varphi, R(\psi')).$$

3. Output $R^{\text{apx}}(\psi')$ together with the distance $\text{MKD}(\varphi, R^{\text{apx}}(\psi'))$.

Theorem 5.8. *Let φ, ψ be two Borel probability measures in $\mathbb{R}^d$, and let $K > 0$. Let r be an MKD_2-reference point with respect to rigid motions and with Lipschitz constant c. Then Algorithm 5.3 finds a rigid motion M^{apx}, such that*
$$\text{MKD}_2(\varphi, M^{\text{apx}}(\psi)) \;\leq\; K + (1+c)\cdot \text{MKD}_2^{\text{opt}}(\varphi, \psi).$$
The runtime is the time to compute the MKD_2 at $O((K^{-1}\cdot m_{p^*}(\psi))^{d-1})$ angles in rotation space, plus the time to compute the reference points.

Proof. Let M^{opt} be the optimal rigid motion minimizing $\text{MKD}_2(\varphi, M(\psi))$ over all rigid motions M. Let $\tau := r(\varphi) - r(M^{\text{opt}}(\psi))$ and $\tau^{\text{ref}} := r(\varphi) - r(\psi)$. Let M^* be a rigid motion minimizing $\text{MKD}_2(\varphi, M(\psi))$ while mapping $r(\psi)$ onto $r(\varphi)$. Let R^{apx} be the approximate rotation determined by Algorithm 5.3.
$$\begin{aligned}
\text{MKD}_2(\varphi, R^{\text{apx}}\circ\tau^{\text{ref}}(\psi)) &\leq K + \min_{R\in\text{Rot}(r(\varphi))} \text{MKD}_2(\varphi, R\circ\tau^{\text{ref}}(\psi))\\
&= K + \text{MKD}_2(\varphi, M^*(\psi))\\
&\leq K + \text{MKD}_2(\varphi, \tau\circ M^{\text{opt}}(\psi))\\
&\leq K + \text{MKD}_2(\varphi, M^{\text{opt}}(\psi)) + \text{MKD}_2(M^{\text{opt}}(\psi), \tau\circ M^{\text{opt}}(\psi))\\
&= K + \text{MKD}_2(\varphi, M^{\text{opt}}(\psi)) + \|\tau\|_2\\
&\leq K + \text{MKD}_2(\varphi, M^{\text{opt}}(\psi)) + c\cdot\text{MKD}_2(\varphi, M^{\text{opt}}(\psi))\\
&= K + (1+c)\cdot\text{MKD}_2(\varphi, M^{\text{opt}}(\psi))
\end{aligned}$$

The runtime of this algorithm is the time to compute the MKD_2-reference points, the time to find the rotation R^{apx}, and the time to compute the MKD_2 between φ and $R^{\text{apx}}\circ\tau^{\text{ref}}(\psi)$. By Theorems 5.6 and 5.7 we know that the time to find R^{apx} equals the time to compute the MKD_2 at most $O((K^{-1}\cdot m_{p^*}(\psi))^{d-1})$ times. □

As before, the runtime can be improved by choosing the minimum of the two first moments. A generalization to arbitrary L_p-distances is also possible.

5.3.4 Similarities

We construct an algorithm for minimizing the MKD under similarities by applying Algorithm 5.3 to the two probability measures, where one of them is scaled by the quotient of the first moments of the two measures. Exchanging the roles of φ and ψ makes a difference in this case. See also the remarks on the abstract approximation algorithm for similarities in Section 1.5.4.

Algorithm 5.4.

1. Compute $r(\varphi)$ and $r(\psi)$ and translate ψ by $r(\varphi) - r(\psi)$.
   Let ψ' be the image of ψ.

2. Determine the first moments $m_{r(\varphi)}(\varphi)$ and $m_{r(\psi')}(\psi')$ and scale ψ' by $m_{r(\varphi)}(\varphi)/m_{r(\psi')}(\psi')$ around the center $r(\varphi)$.
   Let ψ'' be the image of ψ' under this scaling.

3. Determine a rotation $R^{\text{apx}} \in \text{Rot}(r(\varphi))$, such that

$$\text{MKD}(\varphi, R^{\text{apx}}(\psi'')) \;\leq\; K + \min_{R \in \text{Rot}(r(\varphi))} \text{MKD}(\varphi, R(\psi'')).$$

4. Output $R^{\text{apx}}(\psi'')$ together with the distance $\text{MKD}(\varphi, R^{\text{apx}}(\psi''))$.

It is straightforward to generalize Lemma 3.8 and Theorem 3.17 to the continuous case. Thereby it is possible to prove an analogous version of Theorem 3.18. Further, analogously to the proof of Theorem 3.20 we observe the following result:

Observation 5.1. *Let φ, ψ be Borel probability measures, and let $K > 0$. Let r be an MKD_2-reference point with respect to similarities and with Lipschitz constant c. Algorithm 5.3 finds a similarity S^{apx}, such that*

$$\text{MKD}_2(\varphi, S^{\text{apx}}(\psi)) \;\leq\; K + 2(1+c) \cdot \text{MKD}_2^{\text{opt}}(\varphi, \psi).$$

The runtime is the time to compute the MKD_2 at most $O((K^{-1} \cdot m_{p^*}(\varphi))^{d-1})$ times plus the time to compute the reference points.

The transformed Borel probability measure is scaled such that its first moment equals the first moment of the fixed measure before finding the approximating rotation. Therefore the runtime depends on the first moment of the fixed measure. Thus, choosing the measure with the smaller first moment to be fixed decreases the runtime, but also changes the result.

Chapter 6

Bottleneck Distance

A well-known and frequently used metric in shape matching is the bottleneck distance for point sets with an equal number of points in each set. The bottleneck distance is the maximum distance of two matched points minimized over all perfect matchings between the two sets. We give a formal definition:

Definition 6.1. Let $\mathcal{P}^{d,n}$ denote the set of d-dimensional point sets with exactly n points. Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\} \in \mathcal{P}^{d,n}$, and let S_n be the group of permutations. We define the bottleneck distance between A and B as

$$D_{\mathcal{B}}^p(A, B) \; := \; \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left\| a_j - b_{\pi(j)} \right\|_p .$$

It is easy to prove that this defines a metric for point sets of equal size in $\mathbb{R}^d$. This distance measure is preferred whenever it is required that each object in one image has to be matched by exactly one object in the other image. See the papers of Efrat, Itai and Katz [25], and Vaidya [49] for further information on the bottleneck distance.

6.1 Related Work and Results

Efrat, Itai and Katz [25] show how to compute the bottleneck distance between two fixed point sets with n points each in the plane in $O(n^{1.5} \log n)$. They also give approximation algorithms to compute the bottleneck distance for fixed sets in higher dimensions. In $\mathbb{R}^3$, a $(1+\varepsilon_1)$-approximation for the Euclidean bottleneck distance can be computed in $O(n^{11/6+\varepsilon_2})$ time. If we use the L_∞-metric as the underlying norm, this runtime can be improved to $O(n^{1.5} \log^d n)$. In arbitrary dimension a $(1 + \varepsilon)$-approximation of $D_{\mathcal{B}}^p$ can be computed in $O(d(1 + \varepsilon^{-1})^d n^{1.5} \log n)$ time. This holds in any dimension and for any $1 \leq p \leq \infty$.

Efrat, Itai and Katz [25] also describe a method to minimize the bottleneck distance under translations. They first give an algorithm with runtime $O(n^5 \log n)$ for the decision problem. A second algorithm is based on ideas for the first algorithm and on the parametric search paradigm of Megiddo [40] as well as Cole's trick [22]. It solves the optimization problem in $O(n^5 \log^2 n)$ time.

These algorithms to compute the bottleneck distance under translations are the best known methods for finding the exact solution. Therefore fast approximation routines are needed. Efrat, Itai and Katz [25] give a $(1+\sqrt{d}\,)$-approximation algorithm for the optimization problem under translations with runtime $O(n^{1.5} \log n)$. As we discuss later in Section 6.4, this algorithm is a special case of our reference point method if we choose the lower left corner of the smallest axis-parallel hyperrectangle enclosing a point set as the reference point. Using the center of mass as the reference point we can improve this result to an approximation factor of 2 without increasing the runtime.

In contrast to the lower left corner, the center of mass is equivariant with respect to affine transformations. This allows us to construct approximation algorithms for wider classes of transformations, like rigid motions and similarities. Recently, Agarwal and Phillips [1] have shown a 2-approximation algorithm for matching point sets under rotations in the plane in time $O(n^{3.5} \text{poly} \log n)$ if the Euclidean distance is chosen as the underlying metric. The main contribution of this chapter is that we improve this runtime significantly using different ideas. We show a $(1+\sqrt{2})$-approximation for the same problem with runtime $O(n^{2.5} \log n)$. Thus, except for the slightly worse approximation factor, we improve the result of Agarwal and Phillips [1] by a factor of $\Omega(n)$. Based on this, we show a $2(1+\sqrt{2})$-approximation for rigid motions and a $4(1+\sqrt{2})$-approximation for similarity transformations with runtime $O(n^{2.5} \log n)$. Later we use a uniformly distributed δ-net of size $O(\varepsilon^{-1/2})$ on the unit circle to bound the approximation ratio to $2+\varepsilon$. We further derive fully polynomial-time approximation schemes (FPTAS) by standard discretization methods for translations and rigid motions. The dependence on ε^{-1} is quadratic in the first case and cubic in the second case.

We also give an exact algorithm to compute the bottleneck distance under rotations around a fixed point. The runtime of this algorithm is $O(n^{5.5} \log n)$.

Based on the fact that $\|a\|_p \leq \sqrt{d}\, \|a\|_2 \leq d\, \|a\|_p$ for every $a \in \mathbb{R}^d$ and $1 \leq p \leq \infty$, we show that the approximation algorithms generalize to arbitrary L_p-distance as the underlying norm. The approximation ratio increases by a factor of at most $\sqrt{d}$.

We can generalize the approximation algorithms for rigid motions and similarities to any dimension ≥ 3. The method is similar to the one which was used by Klein and Veltkamp [38] for the Earth Mover's Distance (EMD). Unfortunately, the runtime and approximation factor are exponential in the dimension.

6.2 Preliminaries

We prove a basic but fundamental result which allows us to use the abstract approximation algorithms introduced in Section 1.5.

Theorem 6.1. *For any point set $A \in \mathcal{P}^{d,n}$ and any translation $\tau \in \mathbb{R}^d$ we have*

$$D^p_{\mathcal{B}}(A, \tau(A)) \;=\; \|\tau\|_p.$$

Proof. In Theorem 6.7 below we show that the center of mass is a $D^p_{\mathcal{B}}$-reference point with Lipschitz constant 1. Thus,

$$D^p_{\mathcal{B}}(A, \tau(A)) \;\geq\; \|C(A) - C(A+\tau)\|_p \;=\; \|\tau\|_p,$$

where $C(A)$ denotes the center of mass of A. Since any point has a distance of $\|\tau\|_p$ to its translated version we have $D^p_{\mathcal{B}}(A, \tau(A)) \geq \|\tau\|_p$ and the lemma is proven. □

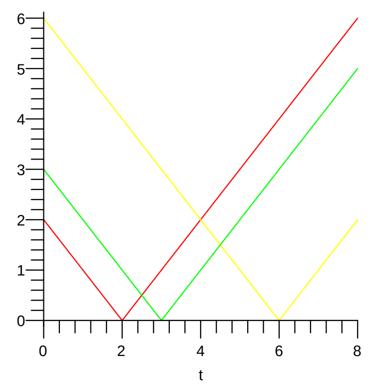


Figure 6.1: Distance Functions Under Translations

6.3 Bottleneck Distance Under Translations on a Line

In general, the problem to exactly compute the minimum bottleneck distance under translations seems to be computationally expensive. In this section we show that this problem is easy if the point sets consist of real numbers, i.e., of points on a line.

Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ be two point sets in $\mathcal{P}^{1,n}$. Let the coordinates a_i and b_j be monotonously increasing. If the point sets are not sorted, we sort them beforehand. This increases the runtime given in Theorem 6.2 to $O(n \log n)$.

In Figure 6.1 we give an example of two point sets $A, B \in \mathcal{P}^{1,3}$ under translation, where $A = \{5, 7, 12\}$ and $B = \{3, 4, 6\}$.

Lemma 6.1. *The minimum bottleneck distance under translations is always induced by the permutation which maps a_j to b_j for every $j = 1, \ldots, d$.*

Proof. We can interpret the matching problem as minimum cost flow problem and use the greedy algorithm given in the proof of Theorem 3.7. □

Theorem 6.2. *The minimum bottleneck distance under translations between sorted point sets on the line can be computed in $O(n)$ time.*

Proof. By Lemma 6.1 we know that

$$D_{\mathcal{B}}^{p\,\mathrm{opt}}(A, B) \;=\; \min_{\tau \in \mathbb{R}} \max_{j=1,\ldots,n} \left\| a_j - b_j - \tau \right\|_p .$$

Hence we have to minimize the upper envelope of n functions describing the distance of points under translation on the line, see Figure 6.1. Let $m := \min_{j=1,\ldots,n}(a_j - b_j)$ and $M := \max_{j=1,\ldots,n}(a_j - b_j)$. The distance function of two points under translation equals the function of the absolute value translated by the difference of the two points. For all these functions we have that the derivative is -1 on the left side of the translation where the two points coincide, and $+1$ on the right. Therefore, the minimum of the upper envelope is assumed by a translation exactly in the middle of the minimum and the maximum difference of two points, i.e., the minimum is attained when $\tau = (M + m)/2$. The optimal distance $D_{\mathcal{B}}^{p\,\mathrm{opt}}(A, B)$ equals $(M - m)/2$. □

6.4 The Lower Left Corner as a Reference Point

Let LL(A) be the point in $\mathbb{R}^d$ where the j-th coordinate of LL(A) is the minimum of all j-th coordinates of all points in A. In $\mathbb{R}^2$ this describes the lower left corner of the smallest axis-parallel rectangle enclosing A. Then the mapping LL: $\mathcal{P}^{d,n} \to \mathbb{R}^d$ is a $D_{\mathcal{B}}^p$-reference point with respect to translations. The ideas involved in the proof have been used by Efrat, Itai and Katz [25] and Alt, Behrends and Blömer [4]. In the first paper [25], an approximation algorithm for the bottleneck distance under translations in $\mathbb{R}^2$ is given, in the second work [4] a similar one is given for the Hausdorff distance, see Section 2.3.1.

Theorem 6.3. *Let $1 \le p \le \infty$. Then the lower left corner* LL: $\mathcal{P}^{d,n} \to \mathbb{R}^d$ *is a $D_{\mathcal{B}}^p$-reference point with respect to translations. Its Lipschitz constant is at most $\sqrt[p]{d}$ for $1 \le p < \infty$ and it is* 1 *for $p = \infty$.*

Proof. The equivariance of the lower left corner under translations is obvious. We show the Lipschitz continuity

$$\|\mathrm{LL}(A) - \mathrm{LL}(B)\|_p \;\le\; \sqrt[p]{d} \cdot D_{\mathcal{B}}^p(A, B).$$

To see this consider

$$\begin{aligned} \|\mathrm{LL}(A) - \mathrm{LL}(B)\|_p &= \sqrt[p]{\sum_{j=1}^{d} |(\mathrm{LL}(A) - \mathrm{LL}(B))_j|^p} \\ &\le \sqrt[p]{d \cdot \max_{j=1,\ldots,d} |(\mathrm{LL}(A) - \mathrm{LL}(B))_j|^p} \\ &= \sqrt[p]{d} \cdot \max_{j=1,\ldots,d} |(\mathrm{LL}(A) - \mathrm{LL}(B))_j|. \end{aligned}$$

Let j^* be an index where the maximum in the above formula is assumed. Let a^* and b^* be points with minimal j^*-th coordinate in A and B, respectively. That is, $|(a^* - b^*)_{j^*}| = \max_{j=1,\ldots,d} |(\mathrm{LL}(A) - \mathrm{LL}(B))_j|$. W.l.o.g. let a^* be the point with the smaller j^*-th coordinate of these two. This point has to be matched to a point of B in every matching and clearly, the distance of a^* to its match is at least the distance of the j^*-th coordinate. Therefore

$$|(a^* - b^*)_{j^*}| \;\le\; \left\|a^* - b_{\pi(a^*)}\right\|_p$$

for every permutation $\pi \in S_n$. Naturally, $\pi(a^*)$ denotes $\pi(k)$, where k is the index of a^*. Since the above inequality holds for any permutation, it especially holds for the minimum, i.e.,

$$|(a^* - b^*)_{j^*}| \;\le\; \min_{\pi \in S_n} \left\|a^* - b_{\pi(a^*)}\right\|_p .$$

The last inequality stays true if we maximize over all distances of all matched points instead of the distance between a^* and $b_{\pi(a^*)}$, i.e.,

$$|(a^* - b^*)_{j^*}| \;\le\; \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left\|a_j - b_{\pi(j)}\right\|_p .$$

This leads to

$$\|\mathrm{LL}(A) - \mathrm{LL}(B)\|_p \leq \sqrt[p]{d} \cdot |(a^* - b^*)_{j^*}| \leq \sqrt[p]{d} \cdot \min_{\pi \in S_n} \max_{j=1,\dots,n} \left\| a_j - b_{\pi(j)} \right\| = \sqrt[p]{d} \cdot D^p_{\mathcal{B}}(A, B).$$

This concludes the proof for $1 \leq p < \infty$. The proof for $p = \infty$ is analogous since by definition $\|\mathrm{LL}(A) - \mathrm{LL}(B)\|_p = \max_{j=1,\dots,d} |(\mathrm{LL}(A) - \mathrm{LL}(B))_j|$. □

The following theorem is a reformulation of a result by Efrat, Itai and Katz [25] in terms of reference points and extended to arbitrary dimension. The proof follows directly, using the result on the abstract approximation Algorithm 1.1 for translations and Theorem 6.1.

Theorem 6.4. *The $D^p_{\mathcal{B}}$-reference point* LL: $\mathcal{P}^{d,n} \to \mathbb{R}^d$ *induces an approximation algorithm for the bottleneck distance under translations with approximation factor $1 + \sqrt[p]{d}$ for $1 \leq p < \infty$ and 2 for $p = \infty$. Its runtime is the time to compute the bottleneck distance in $\mathbb{R}^d$.*

Efrat, Itai and Katz [25] use an ε-grid in translation space around the translation for which the two lower left corners coincide to construct a $(1+\varepsilon)$-approximation algorithm. In arbitrary dimension d, this algorithm has a runtime of $O(\varepsilon^{-2d} n^{1.5} \log n \log \varepsilon^{-1})$. We describe the method in Section 6.8 and generalize the result to arbitrary reference points.

In the following section we give a lower bound for the matching algorithm using the lower left corner in the Euclidean case, i.e., we give an example of two point sets in $\mathbb{R}^2$ where an approximation ratio of $1 + \sqrt{2}$ is assumed. Recalling Theorem 1.2 and the fact that every fixed corner of the smallest axis-parallel hyperrectangle enclosing a point set is a reference point as well, one might think about using a convex combination of all those points as a reference point. A natural candidate is the convex combination defining the center of the set, i.e., the combination where every corner contributes the same part.

Definition 6.2. Let $A \in \mathcal{P}^{d,n}$ be a point set. We define $\mathrm{CC}(A)$ as the sum of all corners of the smallest axis-parallel hyperrectangle enclosing A divided by 2^n.

By Theorem 1.2, CC is a reference point with Lipschitz constant at most $\sqrt[p]{d}$ and therefore leads to a constant-factor approximation of ratio $1 + \sqrt[p]{d}$. Unfortunately, even for this reference point we were able to find a lower bound in the Euclidean case, as we describe in Section 6.4.2. Note that CC is not equivariant under rotations and therefore is a reference point with respect to translations only.

6.4.1 Lower Bound for the Lower Left Corner

We show that the constants given in Theorems 6.3 and 6.4 cannot be improved. We do this for the Euclidean case in the plane. The lower bound is given by two sets where matching of the two lower left corners leads to an approximation factor of $1 + \sqrt{2}$. The lower bound for the Lipschitz constant of LL as a reference point follows from this. Let $A, B \in \mathcal{P}^{2,4}$,

$$\begin{aligned} A &:= \left\{ (-1, 0), \left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right), (0, -1), \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right) \right\}, \\ B &:= \{(0,0), (0,0), (0,0), (0,0)\}. \end{aligned}$$

See Figure 6.2 for an illustration. The two point sets are obviously in optimal position with respect to translations. It follows that $D^{\mathrm{opt}}_{\mathcal{B}}(A, B) = 1$.

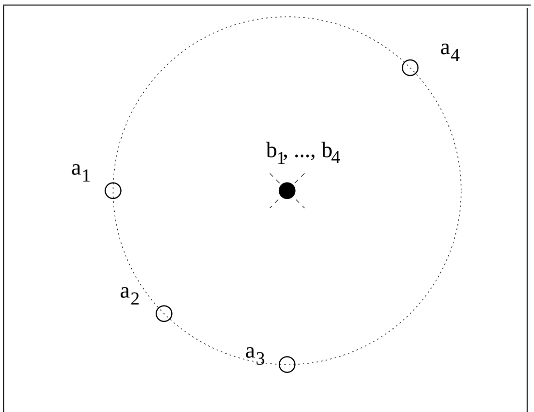


Figure 6.2: Optimal Matching

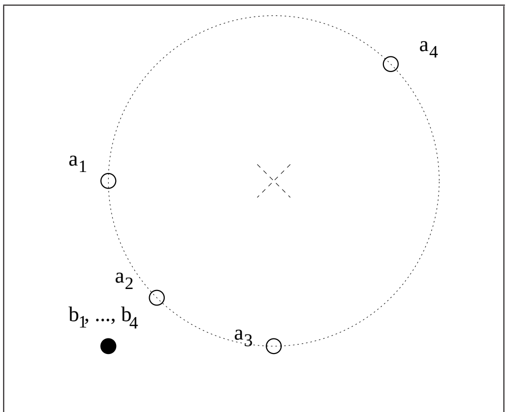


Figure 6.3: LL-Matching

On the other hand, matching with respect to coinciding lower left corners leads to a position where the points of B are placed in $(-1,-1)$, see Figure 6.3. The maximum distance is assumed by any point in B and a_4 and equals $\|b_1 - a_4\|_2 = \sqrt{2(1+\frac{1}{\sqrt{2}})^2} = 1+\sqrt{2}$.

A lower bound in general position can be easily constructed by perturbing B slightly.

6.4.2 Lower Bound for the Center

Similar to the last section we show a lower bound for the approximation algorithm induced by the center of the point set, i.e., the convex combination of all corners of the smallest axis-parallel hyperrectangle enclosing the point set, where any of these points contributes the same part. Again we prove this for the Euclidean case in the plane. By Theorem 1.2 this mapping is a reference point with respect to translations and with Lipschitz constant $\sqrt{2}$. Let the point sets A and B in $\mathcal{P}^{2,7}$ be given by

$$\begin{aligned} A &:= \left\{(0,1),(-1,0),(0,-1),(1,0),\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right),\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right),\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)\right\}, \\ B &:= \{(1,-1),(-1,1),(-1,-1),(-1,-1),(-1,-1),(-1,-1),(-1,-1)\}. \end{aligned}$$

See Figure 6.5 for an illustration.

In the given position $\mathrm{CC}(A) = \mathrm{CC}(B) = (0,0)$ and therefore the approximation algorithm leaves the point sets unchanged. The bottleneck distance in this position is $1+\sqrt{2}$, since at least one point of a_5, a_6 and a_7 has to be matched to some point of $\{b_3,\dots,b_7\}$. But, as can be seen in Figure 6.4, a translation of B by $(1,1)$ leads to a bottleneck distance of 1, proving an approximation ratio of $1+\sqrt{2}$. The lower bound for the Lipschitz constant of the center as a reference point follows from this.

Again, a lower bound using point sets in general position can be constructed by perturbing the two point sets slightly.

6.5 Center of Mass of the Boundary of the Convex Hull

Alt, Behrends and Blömer [4] show that the center of mass of the boundary of the convex hull of a compact subset in the plane is a reference point for the Hausdorff distance with respect

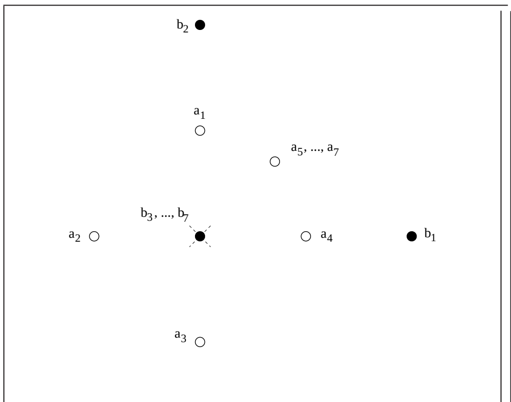


Figure 6.4: Optimal Matching

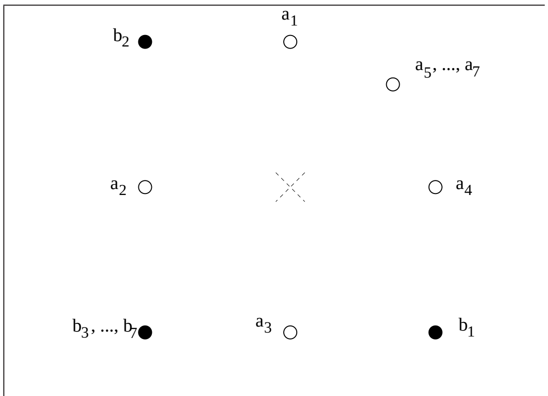


Figure 6.5: CC-Matching

to similarities, see Chapter 2. They do this in the case of $\mathbb{R}^2$ equipped with the Euclidean distance. This is the first mapping shown to be a reference point with respect to similarities. Its Lipschitz constant is at most $4\pi + 3$. We show that the center of mass of the boundary of the convex hull is also a $D^2_{\mathcal{B}}$-reference point.

We first prove that the Hausdorff distance $D^2_{\mathcal{H}}$ is a lower bound for the bottleneck distance.

Lemma 6.2. *Let $A, B \in \mathcal{P}^{d,n}$. Then*

$$D^2_{\mathcal{H}}(A, B) \;\leq\; D^2_{\mathcal{B}}(A, B).$$

Proof. Let $\pi^* \in S_n$ be a permutation inducing the bottleneck distance between A and B. Let $\vec{D}^2_{\mathcal{H}}(A, B)$ denote the directed Hausdorff distance from A to B. Then,

$$\begin{aligned} D^2_{\mathcal{B}}(A, B) &= \max_{j=1,\ldots,n} \left\| a_j - b_{\pi^*(j)} \right\|_2 \\ &\geq \max_{j=1,\ldots,n} \min_{i=1,\ldots,n} \left\| a_j - b_i \right\|_2 \\ &= \vec{D}^2_{\mathcal{H}}(A, B). \end{aligned}$$

Using $D^2_{\mathcal{B}}(A, B) = D^2_{\mathcal{B}}(B, A)$, an analogous proof shows $D^2_{\mathcal{B}}(A, B) \geq \vec{D}^2_{\mathcal{H}}(B, A)$. □

Using Theorem 1.3, we obtain the following result:

Theorem 6.5. *The center of mass of the boundary of the convex hull is a $D^2_{\mathcal{B}}$-reference point with respect to similarities. Its Lipschitz constant is at most $4\pi + 3$.*

In contrast, the center of mass of the volume of the convex hull and the center of mass of the vertices of the convex hull are not $D^2_{\mathcal{B}}$-reference points.

Again, we obtain an approximation algorithm for the bottleneck distance under translations by Theorem 6.1 and the abstract approximation Algorithm 1.1.

Theorem 6.6. *The center of mass of the boundary of the convex hull as a $D^2_{\mathcal{B}}$-reference point induces an approximation algorithm for the bottleneck distance under translations with approximation factor $4\pi + 4$. The runtime of this algorithm is $O(n^{1.5} \log n)$.*

Proof. The time to compute the convex hull is $O(n \log n)$ and the center of mass can be computed in linear time. Therefore, the runtime of the algorithm is dominated by the time needed to compute the bottleneck distance in the plane, which is $O(n^{1.5} \log n)$. □

6.6 The Center of Mass as a Reference Point

In this section we show that the center of mass $C(A) = \frac{1}{n}\sum_{j=1}^{n} a_j$ of a point set $A = \{a_1, \ldots, a_n\} \in \mathcal{P}^{d,n}$ is a $D_{\mathcal{B}}^p$-reference point with respect to affine transformations.

Theorem 6.7. *Let $1 \leq p \leq \infty$. The center of mass $C\colon \mathcal{P}^{d,n} \to \mathbb{R}^d$ is a $D_{\mathcal{B}}^p$-reference point with respect to affine transformations. Its Lipschitz constant is* 1.

Proof. The equivariance of the center of mass of a point set under affine transformations is well-known. We prove the Lipschitz continuity as follows:

Let $A = \{a_1, \ldots, a_n\}, B = \{b_1, \ldots, b_n\} \in \mathcal{P}^{d,n}$ be two point sets in $\mathbb{R}^d$. Let $\pi^* \in S_n$ be a permutation inducing the bottleneck distance. Then,

$$\begin{aligned}
\|C(A) - C(B)\|_p &= \left\| \frac{1}{n}\sum_{j=1}^{n} a_j - \frac{1}{n}\sum_{j=1}^{n} b_j \right\|_p \\
&= \left\| \frac{1}{n}\sum_{j=1}^{n} a_j - \frac{1}{n}\sum_{j=1}^{n} b_{\pi^*(j)} \right\|_p \\
&\leq \frac{1}{n}\sum_{j=1}^{n} \left\| a_j - b_{\pi^*(j)} \right\|_p \\
&\leq \frac{1}{n} n \cdot \max_{j=1,\ldots,n} \left\| a_j - b_{\pi^*(j)} \right\|_p \\
&= D_{\mathcal{B}}^p(A, B).
\end{aligned}$$

This proves that the Lipschitz constant of the center of mass as a $D_{\mathcal{B}}^p$-reference point is at most 1. The lower bound follows directly by the fact that the Lipschitz constant of any reference point is at least 1, see Theorem 1.4. □

The last proof does not make any use of the properties of the L_p-metric. In fact, the center of mass is a reference point for the bottleneck distance defined on any norm on the underlying space $\mathbb{R}^d$. The Lipschitz constant in all of these cases is 1.

Using results of Rubner, Tomasi and Guibas [44], and Cohen [19], it was shown by Klein and Veltkamp [37] that the center of mass is a reference point for the Earth Mover's Distance, see Chapter 3. Observing basic network flow theory, in the case of two point sets with the same number of points and equal weight in each point, there is always a minimum cost flow inducing the EMD which also induces a perfect matching between the two sets. Since the bottleneck distance is the maximum distance of a matched pair of points and the EMD is the average distance, we have that $\text{EMD}(A, B) \leq D_{\mathcal{B}}(A, B)$ for all point sets $A, B \in \mathcal{P}^{d,n}$. Thus, Theorem 6.7 follows directly by Theorem 1.3 and the fact that the center of mass is an EMD-reference point with Lipschitz constant 1.

Again we apply the abstract approximation Algorithm 1.1. Using Theorems 1.5 and 6.1 we obtain the following result:

Theorem 6.8. *Let $1 \leq p \leq \infty$. The center of mass as a $D_{\mathcal{B}}^p$-reference point induces an approximation algorithm for the bottleneck distance under translations with approximation factor* 2. *Its runtime is the time to compute the bottleneck distance in $\mathbb{R}^d$.*

6.6.1 Lower Bound for the Center of Mass

In this section we prove a lower bound for the approximation algorithm for the bottleneck distance under translations using the center of mass as a $D^p_{\mathcal{B}}$-reference point. The lower bound does not equal 2, but tends to 2 as the number of used points tends to infinity. Consider the following two point sets $A, B \in \mathcal{P}^{d,n}$:

- A consists of the point a located at coordinates $(2, 0)$ and $n - 1$ points located at the origin.
- B is be a set of n points located at $(1, 0)$.

Obviously, A and B are in optimal position with respect to translations and the bottleneck distance between the two sets is 1. For n tending to infinity, the center of mass of A tends to the origin. The center of mass of B stays at $(1, 0)$. Therefore, matching in a way that the two centers of mass coincide translates B in a way that all its points are located near the origin. One point of B has to be matched to $a = (2, 0)$ and thus, the length of the longest matching arc tends to 2 as n tends to infinity.

6.7 The Steiner Point as a Reference Point

Alt, Aichholzer and Rote [3] show that the Steiner point is a reference point for the Hausdorff distance for compact subsets of $\mathbb{R}^d$ with respect to similarities, see Section 2.3.3. They consider the case of $\mathbb{R}^d$ equipped with the Euclidean distance. Therefore we also restrict our considerations to this case. The Steiner point is a $D^2_{\mathcal{B}}$-reference point with respect to similarities. By Theorem 1.3, this is a direct consequence of the fact that $D^2_{\mathcal{H}}(A, B) \leq D^2_{\mathcal{B}}(A, B)$ for all point sets $A, B \in \mathcal{P}^{d,n}$, see Lemma 6.2. Its Lipschitz constant is

$$\chi_d \;=\; \frac{2d \cdot \mathrm{Vol}(\mathbb{B}^{d-1})}{\mathrm{Vol}(\mathbb{S}^{d-1})} \;=\; \frac{2\Gamma(d/2+1)}{\sqrt{\pi}\,\Gamma(d/2+1/2)},$$

where $\mathbb{B}^d$ denotes the d-dimensional unit ball and $\mathbb{S}^{d-1}$ its boundary, the $(d-1)$-dimensional unit sphere in $\mathbb{R}^d$, and Γ denotes the Gamma function.

Theorem 6.9. *The Steiner point is a $D^2_{\mathcal{B}}$-reference point with respect to similarities. Its Lipschitz constant is χ_d, which for $d = 2$ is $4/\pi$, for $d = 3$ is $3/2$, and for arbitrary dimension d lies between $\sqrt{2/\pi}\sqrt{d}$ and $\sqrt{2/\pi}\sqrt{d+1}$.*

Again we obtain an approximation algorithm for the bottleneck distance under translations using Theorems 1.5 and 6.1.

Theorem 6.10. *The Steiner point as a $D^2_{\mathcal{B}}$-reference point induces an approximation algorithm for the bottleneck distance under translations with approximation factor $1+\chi_d$. The runtime of this algorithm is $O(n^{1.5} \log n)$ in the plane. In general, the runtime equals the time to compute the Steiner points plus the time to compute the bottleneck distance in $\mathbb{R}^d$.*

Proof. The runtime of the algorithm is the sum of the time to compute the Steiner points and the time to compute the bottleneck distance. In the plane, the Steiner point of a point set is defined as the Steiner point of the convex hull, see Alt, Aichholzer and Rote [3]. Computing the convex hull of a point set in the plane takes $O(n \log n)$ time and the Steiner point can be computed in linear time. Therefore, the runtime of the algorithm in the plane is dominated by the time needed to compute the bottleneck distance, which is $O(n^{1.5} \log n)$. □

6.8 FPTAS for Translations

Efrat, Itai and Katz [25] use the constant-factor approximation induced by the lower left corner, see Section 6.4, to obtain a $(1+\varepsilon)$-approximation algorithm for the bottleneck distance under translations. In Section 1.5.2 we have shown how to generalize their approach to arbitrary reference points and to a huge class of distance measures on shapes. Thus, the following result is a direct consequence of Theorems 1.6 and 6.1. We further apply the $(1+\varepsilon)$-approximation for the bottleneck distance between fixed sets with a runtime of $O(d(1+\varepsilon^{-1})^d n^{1.5} \log n)$ by Efrat, Itai and Katz [25].

Theorem 6.11. *[25] Let $A, B \in \mathcal{P}^{d,n}$ and let r be a $D^p_{\mathcal{B}}$-reference point with respect to translations and with Lipschitz constant c. There exists an algorithm that for any $\varepsilon > 0$ finds in time $O(\varepsilon^{-2d} n^{1.5} \log n \log \varepsilon^{-1})$ a translation τ^ε such that*

$$D^p_{\mathcal{B}}(A, B + \tau^\varepsilon) \;\le\; (1+\varepsilon) \cdot D^p_{\mathcal{B}}(A, B + \tau^{\text{opt}}).$$

In the plane we can compute the optimal distance in $O(n^{1.5} \log n)$ time at each of the $O(\varepsilon^{-2})$ grid points and get a $(1+\varepsilon)$-approximation in $O(\varepsilon^{-2} n^{1.5} \log n)$ time.

6.9 Exact Algorithm for Rotations in the Plane

We give a polynomial time algorithm to compute the minimum bottleneck distance under rotations around a fixed point in the plane. We do this for the bottleneck distance based on L_p-distances where p is a rational number. The question if there is a polynomial algorithm if p is irrational is open. Thus for the whole section we assume that $1 \le p \le \infty$ and that p is rational if $\ne \infty$. We further assume that p is a constant.

We first discuss how the distance of two points in the plane changes while one of them is rotated around a fixed rotation center. W.l.o.g. we fix the rotation center to the origin. Let $a := (a_x, a_y)$ be the unrotated point and let $b := (b_x, b_y)$. The distance of a and b under a counterclockwise rotation of b by an angle β is

$$\|a - R_\beta(b)\|_p \;=\; \sqrt[p]{|a_x - b_x \cdot \cos\beta + b_y \cdot \sin\beta|^p + |a_y - b_x \cdot \sin\beta - b_y \cdot \cos\beta|^p}$$

if $1 \le p < \infty$ and

$$\|a - R_\beta(b)\|_\infty \;=\; \max\{|a_x - b_x \cdot \cos\beta + b_y \cdot \sin\beta|\,, |a_y - b_x \cdot \sin\beta - b_y \cdot \cos\beta|\}$$

for $p = \infty$.

Lemma 6.3. *Let $f_1 \ne f_2$ be two functions describing the distance of two pairs of points under rotations around the same center. The number of angles β where $f_1(\beta) = f_2(\beta)$ is bounded by a constant.*

Proof. It is well-known that we can parameterize the sine and cosine as

$$\sin\beta \;=\; \frac{2t}{1+t^2} \quad \text{and} \quad \cos\beta \;=\; \frac{1-t^2}{1+t^2}.$$

In each of the functions f_1, f_2 there are two absolute values, which leads to 4 possible sign patterns in each of the functions. Thus, checking the equality of f_1 and f_2 can be done by checking the equality for the at most 16 sign patterns of the two functions. Since $p \in \mathbb{Q}$ if $\neq \infty$, every equation is algebraic and of constant degree. Therefore this equation has at most a constant number of solutions and thus the overall number of angles β where $f_1(\beta) = f_2(\beta)$ is bounded by a constant. □

Since p is a constant, every distance function as described above is algebraic and of constant degree. We assume that we can minimize those functions on an arbitrary interval in constant time. Further we assume that we can compute all angles where two of these distance functions are equal in constant time.

Note that it is open if there is a global bound on the number of solutions which is independent of p. Using this lemma we can prove the following theorem:

Theorem 6.12. *Let $A, B \in \mathcal{P}^{2,n}$ be two planar point sets and let $p^* \in \mathbb{R}^2$ be a fixed point. Let $1 \leq p \leq \infty$ and $p \in \mathbb{Q}$ if $\neq \infty$. We can compute the minimum bottleneck distance $D^p_{\mathcal{B}}$ under rotations of B around p^* in $O(n^{5.5} \log n)$ time.*

Proof. Let π be a fixed permutation. If π induces the minimum bottleneck distance under rotations of B around p^*, we can write

$$\min_{R \in \mathrm{Rot}(p^*)} D^p_{\mathcal{B}}(A, R(B)) \;=\; \min_{\phi \in [0,2\pi]} \max_{j=1,\dots,n} \left\| a_j - R_\phi(b_{\pi(j)}) \right\|_p .$$

Therefore we can determine the optimal angle by minimizing the upper envelope of n distance functions as described above.

In the following we prove that there are $O(n^4)$ angles where the permutation set inducing the bottleneck distance can change. For $i = 1, 2$ let ϕ_i be a fixed angle and let π_i be a permutation inducing the bottleneck distance of A and $R_{\phi_i}(B)$. Consider the functions

$$f_{\pi_i} \colon [0, 2\pi] \to \mathbb{R}_{\geq 0}, \quad \phi \mapsto \max_{j=1,\dots,n} \left\| a_j - R_\phi(b_{\pi_i(j)}) \right\|_p .$$

Since f_{π_i} is the upper envelope of n continuous functions, f_{π_i} is continuous itself. Assume π_1 does not induce the bottleneck distance between A and $R_{\phi_2}(B)$, and π_2 does not induce the bottleneck distance between A and $R_{\phi_1}(B)$. This implies that $f_{\pi_1}(\phi_1) < f_{\pi_2}(\phi_1)$ and $f_{\pi_1}(\phi_2) > f_{\pi_2}(\phi_2)$. Since f_{π_1} and f_{π_2} are continuous, there exists an angle ϕ^*, such that $f_{\pi_1}(\phi^*) = f_{\pi_2}(\phi^*)$. This implies that

$$\max_{j=1,\dots,n} \left\| a_j - R_{\phi^*}(b_{\pi_1(j)}) \right\|_p \;=\; \max_{j=1,\dots,n} \left\| a_j - R_{\phi^*}(b_{\pi_2(j)}) \right\|_p .$$

Thus there are points $a_1, a_2 \in A$ and $b_1, b_2 \in B$ such that $\|a_1 - R_{\phi^*}(b_1)\|_p = \|a_2 - R_{\phi^*}(b_2)\|_p$. There are $O(n^2)$ pairs of points in A and B. Therefore the number of pairs of pairs of points is $O(n^4)$. The distance of a fixed pair of points equals the distance of another pair of points at at most a constant number of rotations, see Lemma 6.3 and the following comments. This directly leads to $O(n^4)$ event angles which can be computed in $O(n^4)$ time. Computing the bottleneck distance at the midpoint between two subsequent event angles determines a permutation inducing the bottleneck distance on the interval between the two event angles. This function can be minimized in constant time on this interval, see the remarks above.

Thus we have shown that the following algorithm computes the optimal bottleneck distance under rotations of B around p^*.

Algorithm 6.1.

1. Compute the $O(n^4)$ event angles and sort them.
2. Let $\phi_1, \dots, \phi_m, \phi_{m+1} := \phi_1$ be the sorted list.
3. For any $i = 1, \dots, m-1$ do:
   (a) Let $\phi := (\phi_{i+1} - \phi_i)/2$.
   (b) Consider a permutation π inducing $D^p_{\mathcal{B}}(A, R_\phi(B)$.
   (c) Determine f_π.
   (d) Minimize f_π on the interval $[\phi_i, \phi_{i+1}]$.
4. Output the global minimum.

The runtime of this algorithm is the time to compute the bottleneck distance at the $O(n^4)$ event angles. Sorting these angles needs $O(n^4 \log n)$ time. Since computing the bottleneck distance once needs $O(n^{1.5} \log n)$ time, the claim follows. □

6.10 Rigid Motions

We now investigate rigid motions and give several algorithms to minimize the bottleneck distance under this class of transformations. As in the corresponding Section 3.4.3 for the EMD we show a first method using an oracle which finds the optimal rotation around a fixed point. In contrast to the EMD, where we do not know an exact algorithm, we have seen a solution for this problem in Section 6.9. This method has a runtime of $O(n^5 \log n)$ and it can be applied for any rational p, where $1 \leq p < \infty$, or $p = \infty$.

6.10.1 Approximation for Rigid Motions

The first algorithm provides a tool for the development of efficient algorithms:

Algorithm 6.2.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.
2. Find an optimal matching of A and B' under rotations of B' around $r(A)$.
   Let B'' be the image of B' under this rotation.
3. Output B'' together with the distance $D^p_{\mathcal{B}}(A, B'')$.

The following theorem is a direct consequence of Theorems 1.7 and 6.1.

Theorem 6.13. *Let $A, B \in \mathcal{P}^{d,n}$. Let r be a $D^p_{\mathcal{B}}$-reference point with respect to rigid motions and with Lipschitz constant c. Algorithm 6.2 finds an approximately optimal matching for rigid motions with approximation factor $c+1$ in time $O(T^{\mathrm{ref}}(n) + T^{D^p_{\mathcal{B}}}(n) + T^{\mathrm{rot}}(n))$.*

All reference points with respect to a transformation class including the set of rigid motions mentioned in this chapter may be applied to the last theorem. Therefore this result leads to several approximation algorithms with different approximation ratios, depending on the Lipschitz constant of the used reference point. For instance, applying the center of mass as a $D^p_{\mathcal{B}}$-reference point leads to the following corollary:

Corollary 6.1. *Algorithm 6.2 using the center of mass as a $D^p_{\mathcal{B}}$-reference point finds an approximately optimal matching for rigid motions with approximation factor 2 in time $O(T^{D^p_{\mathcal{B}}}(n) + T^{\text{rot}}(n))$.*

Lower Bound for Algorithm 6.2

A lower bound for Algorithm 6.2 using the center of mass as $D^p_{\mathcal{B}}$-reference point is given by the same sets used in Section 6.6.1. These point sets are in optimal position with respect to rotations. Therefore an approximation ratio tending to 2 is assumed as n tends to infinity.

6.10.2 Rigid Motion Approximation Using Rotation Approximation

We use Algorithm 6.2 to construct faster approximation algorithms for rigid motions. Similar to the corresponding Section 3.4.4 for the EMD we prove a lemma that suggests a constant-factor approximation for the bottleneck distance under rotations around a fixed point. We prove the result for the Euclidean case first and generalize it to arbitrary L_p-distance later. The proof of the following lemma is based on a result by Giannopoulos [26].

Lemma 6.4. *Let $A, B \in \mathcal{P}^{d,n}$ be two point sets and let $p^* \in \mathbb{R}^d$ be a fixed point. Let $\text{Rot}(p^*)$ be the set of rotations around p^*. There is a rotation $R^{\text{apx}} \in \text{Rot}(p^*)$ such that*

$$D^2_{\mathcal{B}}(A, R^{\text{apx}}(B)) \;\leq\; 2 \cdot \min_{R \in \text{Rot}(p^*)} D^2_{\mathcal{B}}(A, R(B)),$$

where R^{apx} aligns p^*, a point of A, and a point of B.

Proof. W.l.o.g. let A and B be in optimal position with respect to rotations of B around p^*. In the proof of Lemma 3.4 we have shown that there is a rotation R^{apx}, which aligns p^*, a point of A, and a point of B, and additionally, for every pair of points $a \in A$ and $b \in B$ we have

$$\|a - R^{\text{apx}}(b)\|_2 \;\leq\; 2\,\|a - b\|_2.$$

This surely holds for any matched pair of points and therefore we conclude

$$\begin{aligned} D^2_{\mathcal{B}}(A, R^{\text{apx}}(B)) &= \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left\|a_j - R^{\text{apx}}(b_{\pi(j)})\right\|_2 \\ &\leq 2 \cdot \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left\|a_j - b_{\pi(j)}\right\|_2 \\ &= 2 \cdot D^2_{\mathcal{B}}(A, B). \end{aligned}$$

□

Analogous to Lemma 3.5 we extend the result to any L_p-distance, where $1 \leq p \leq \infty$:

Lemma 6.5. *Let $A, B \in \mathcal{P}^{d.n}$ be two point sets and let $p^* \in \mathbb{R}^d$ be a fixed point. Let $\text{Rot}(p^*)$ be the set of rotations around p^*. There exists a rotation $R^{\text{apx}} \in \text{Rot}(p^*)$ such that*

$$D^p_{\mathcal{B}}(A, R^{\text{apx}}(B)) \leq 2\sqrt{d} \cdot \min_{R \in \text{Rot}(p^*)} D^p_{\mathcal{B}}(A, R(B)),$$

where R^{apx} aligns p^*, a point of A, and a point of B.

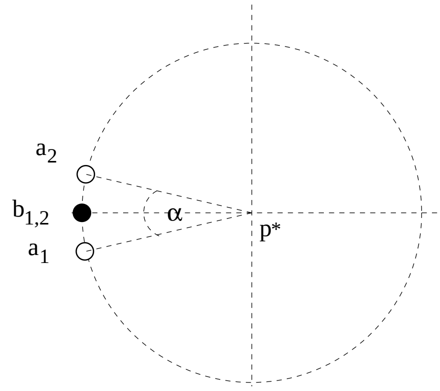


Figure 6.6: Lower Bound for Lemma 6.4.

Lower Bound for Lemma 6.4

A lower bound for the approximation constant given in Lemma 6.4 is easy to find, see Figure 6.6. The depicted constellation describes the minimum bottleneck distance under rotations around p^*. The approximation found in Lemma 6.4 rotates the point set B by either $\alpha/2$ or $-\alpha/2$. For α tending to 0 this leads to a 2-approximation.

Approximation for Rigid Motions in the Plane

As in Section 3.4.4 we use Lemmas 6.4 and 6.5 to construct an approximation algorithm for the bottleneck distance under rigid motions in the plane. The routine is based on Algorithm 6.2. The difficult part in this algorithm, namely to find the optimal rotation around a fixed point, is substituted by the following method:

Algorithm 6.3.

1. Compute the minimum $D^p_{\mathcal{B}}$ over all possible alignments of the coinciding reference points, a point of A, and a point of B.

Since there are $O(n^2)$ possibilities to align the reference point and two points of A and B, the runtime is $O(n^2 \cdot T^{D^p_{\mathcal{B}}}(n))$. Using this algorithm we obtain an easy to implement approximation for rigid motions.

Algorithm 6.4.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.

2. Find an optimal matching of A and B' under rotations of B' around $r(A)$, where $r(A)$, a point of A, and a point of B are aligned.
   Let B'' be the image of B' under this rotation.

3. Output B'' together with the distance $D^p_{\mathcal{B}}(A, B'')$.

Using the above remarks, the following theorem is a direct consequence of the results on our abstract algorithm for rigid motions, see Theorem 1.7, as well as Theorem 6.1 and Lemma 6.4.

Theorem 6.14. *Let $A, B \in \mathcal{P}^{2,n}$ be planar point sets. Let r be a $D^p_{\mathcal{B}}$-reference point with respect to rigid motions and with Lipschitz constant c. Algorithm 6.4 finds an approximately optimal matching for rigid motions with approximation factor $2(c+1)$ in time $O(T^{\mathrm{ref}}(n) + n^{3.5}\log n)$.*

The proof of the following theorem is analogous to the proof of Theorem 6.14, using Lemma 6.5 instead of Lemma 6.4.

Theorem 6.15. *Let $1 \leq p \leq \infty$ and let $A, B \in \mathcal{P}^{2,n}$ be planar point sets. Let r be a $D^p_{\mathcal{B}}$-reference point with respect to rigid motions and with Lipschitz constant c. Algorithm 6.4 finds an approximately optimal matching for $D^p_{\mathcal{B}}$ under rigid motions in the plane with approximation factor $2\sqrt{2}\,(c+1)$ in time $O(T^{\mathrm{ref}}(n) + n^{3.5}\log n)$.*

In the following corollary we apply the center of mass to the last two theorems:

Corollary 6.2. *Algorithm 6.4 using the center of mass as a $D^p_{\mathcal{B}}$-reference point induces an approximation algorithm with approximation factor 4 in the case of the Euclidean distance and $4\sqrt{2}$ for arbitrary L_p-distance, where $1 \leq p \leq \infty$. Its runtime is $O(n^{3.5}\log n)$ in the Euclidean case and $O(n^2 \cdot T^{D^p_{\mathcal{B}}}(n))$ for arbitrary $1 \leq p \leq \infty$.*

Approximation for Rigid Motions in Higher Dimensions

It is possible to generalize Algorithm 6.4 to dimensions ≥ 3. The proofs of the following two theorems are analogous to those of the corresponding Theorems 3.15 and 3.16 for the EMD and are omitted.

Theorem 6.16. *Let $A, B \in \mathcal{P}^{d,n}$, where $d \geq 3$. Let r be a $D^2_{\mathcal{B}}$-reference point with respect to rigid motions and with Lipschitz constant c. An approximate rigid motion M^{apx} such that*

$$D^2_{\mathcal{B}}(A, M^{\mathrm{apx}}(B)) \;\leq\; 2^{d-1}(c+1)\cdot D^{2\,\mathrm{opt}}_{\mathcal{B}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(n) + n^{2(d-1)} \cdot T^{D^2_{\mathcal{B}}}(n))$ time.

Using Lemma 6.5 we can extend the result to any L_p-norm:

Theorem 6.17. *Let $A, B \in \mathcal{P}^{d,n}$, where $d \geq 3$, and let $1 \leq p \leq \infty$. Let r be a $D^p_{\mathcal{B}}$-reference point with respect to rigid motions and with Lipschitz constant c. An approximate rigid motion M^{apx} such that*

$$D^p_{\mathcal{B}}(A, M^{\mathrm{apx}}(B)) \;\leq\; 2^{d-1}\sqrt{d}\,(c+1)\cdot D^{p\,\mathrm{opt}}_{\mathcal{B}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(n) + n^{2(d-1)} \cdot T^{D^p_{\mathcal{B}}}(n))$ time.

We apply the center of mass to the last two theorems:

Corollary 6.3. *The center of mass as a $D^p_{\mathcal{B}}$-reference point induces an approximation algorithm with approximation factor 2^d in the case of the Euclidean distance and $2^d\sqrt{d}$ for arbitrary L_p-distance, where $1 \leq p \leq \infty$. The runtime is $O(n^{2(d-1)} \cdot T^{D^p_{\mathcal{B}}}(n))$.*

6.10.3 Rigid Motion Approximation - An Improved Version

We present an improvement on Algorithm 6.4 to compute the bottleneck distance under rigid motions. Using Algorithm 6.4 in the plane, we have to compute the bottleneck distance whenever at least one point of each set and the rotation center are aligned, that is, $O(n^2)$ times. A different approach but also leading to $O(n^2)$ point-to-point correspondences is given by Agarwal and Phillips [1]. We reduce this number to $O(n)$ events by observing that it suffices to consider those alignments where the furthest point from the rotation center is aligned with some point of the other set. If there is more than one point at the same furthest distance from the rotation center it suffices to arbitrarily choose one. We have to pay for the decreased runtime by a slightly worse approximation ratio of $1+\sqrt{2}$ against 2. However, later we use a uniformly distributed δ-net of size $O(\varepsilon^{-1/2})$ on the unit circle to bound the approximation ratio to $2+\varepsilon$, see Theorem 6.8.

If not stated otherwise we assume that one of the furthest points is in B. If not, we can exchange the roles of A and B, which does not make a difference if only rigid motions are under consideration. We start with the $(1+\sqrt{2}\,)$-approximation in the Euclidean case.

Lemma 6.6. *Let $A, B \in \mathcal{P}^{d,n}$ be two point sets in dimension $d \geq 2$ and let S^* be a fixed d'-dimensional affine space with $0 \leq d' < d-1$. Let $A \setminus S^* \neq \emptyset$ and $B \setminus S^* \neq \emptyset$. Let $\mathrm{Rot}(S^*)$ be the set of rotations leaving S^* invariant. Let $b_1 \in B$ be a point among all points of both sets with maximum distance to S^*. There exists a rotation $R' \in \mathrm{Rot}(S^*)$ such that*

$$D^2_{\mathcal{B}}(A, R'(B)) \;\leq\; (1+\sqrt{2}\,) \cdot \min_{R \in \mathrm{Rot}(S^*)} D^2_{\mathcal{B}}(A, R(B)),$$

where R' rotates B such that S^*, $R'(b_1)$ and some point of $A \setminus S^*$ are in a $(d'+1)$-dimensional space.

Proof. W.l.o.g. let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ be in optimal position with respect to rotations of B around S^*. After renumbering we can assume that there is a permutation inducing $D^2_{\mathcal{B}}(A, B)$ and mapping a_j to b_j for all $j = 1, \ldots, n$. Let $p^* \in S^*$ be a fixed point. Let $S^\perp := p^* + (S^*)^\perp$ be the orthogonal affine space containing p^*. For all points $a_j \in A \setminus S^*$ let $a_j^\perp$ be its orthogonal projection onto $S^\perp$. Let $b_1^\perp$ be the orthogonal projection of b_1 onto $S^\perp$. For every $1 \leq j \leq n$ let R_j be the rotation with the following properties:

1. R_j rotates around the $(d-2)$-dimensional subspace S_j that contains S^* and is orthogonal to the plane E_j spanned by the points p^*, $a_j^\perp$ and $b_1^\perp$.
2. $R_j(B)$ aligns p^*, $a_j^\perp$ and $b_1^\perp$.
3. For the rotation angle $\phi(R_j)$ we have $|\phi(R_j)| \leq \pi$.

Note that if $\phi(R_j) = \pi$ the plane E_j is not uniquely defined in property 1. In this case we can use any plane orthogonal to S^* containing the points p^*, $a_j^\perp$ and $b_1^\perp$. Further note that the rotation angle is given by the smaller angle between the lines through $a_j^\perp$ and p^*, and $b_1^\perp$ and p^*. Thus this angle is defined in arbitrary dimension. Furthermore, R_j is independent of the concrete choice of $p^* \in S^*$, since choosing a different point only causes a parallel translation of $a_j^\perp$ and $b_1^\perp$. Let $\mathcal{R}$ be the set of all rotations R_j. Let $R' \in \mathcal{R}$ such that for all $R \in \mathcal{R}$ we have $|\phi(R')| \leq |\phi(R)|$. Let $\phi' := \phi(R')$. If $\phi' = 0$, R' fulfills the claim of the lemma.

Let a_k be a point inducing $R' = R_k$ and let a_i and b_i be two arbitrary points. Let a_i' and b_i' be the orthogonal projections of a_i and b_i onto E_k, respectively. Let $\overline{a_i} := a_i - a_i'$ and

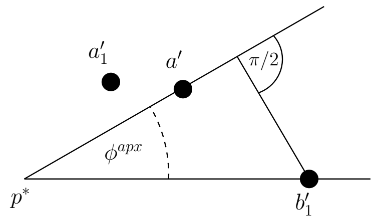


Figure 6.7: $\sin(\phi') \cdot \|b'_1 - p^*\|_2 \leq \|b'_1 - a'_1\|_2$

$\overline{b_i} := b_i - b'_i$. Now, $a_i = a'_i + \overline{a_i}$ and $b_i = b'_i + \overline{b_i}$. Since $\overline{b_i}$ is parallel to S_k, this vector is invariant under R' and therefore we have $R'(b_i) = R'(b'_i + \overline{b_i}) = R'(b'_i) + \overline{b_i}$.

We prove the result by a case distinction on the angle ϕ':

1. Let $\phi' \in (0, \pi/2]$. We first bound the distance between the projections of two matched points inside the plane E_k after the rotation by R'. For every matched pair, i.e., for every $j = 1, \ldots, n$ we have

$$\begin{aligned}
&\left\|a'_j - R'(b'_j)\right\|_2 \\
&\leq \left\|a'_j - b'_j\right\|_2 + \left\|b'_j - R'(b'_j)\right\|_2 \\
&\leq \left\|a'_j - b'_j\right\|_2 + 2 \cdot \sin(\phi'/2) \cdot \left\|b'_j - p^*\right\|_2 \\
&\leq \left\|a'_j - b'_j\right\|_2 + 2 \cdot \sin(\phi'/2) \cdot \left\|b'_1 - p^*\right\|_2, \quad \text{by assumption} \\
&\leq \left\|a'_j - b'_j\right\|_2 + \sqrt{2} \cdot \sin(\phi') \cdot \left\|b'_1 - p^*\right\|_2, \quad \text{since } \tfrac{\sin(\phi'/2)}{\sin(\phi')} = \tfrac{1}{2\cos(\phi'/2)} \leq 1/\sqrt{2} \\
&\leq \left\|a'_j - b'_j\right\|_2 + \sqrt{2} \cdot \left\|a'_1 - b'_1\right\|_2, \quad \text{see Figure 6.7.}
\end{aligned}$$

We now bound the maximum distance after the rotation by R' between all pairs of matched points:

$$\begin{aligned}
&\|a_j - R'(b_j)\|_2 \\
&= \left\|a'_j + \overline{a_j} - R'(b'_j) - \overline{b_j}\right\|_2 \\
&= \sqrt{\left\|a'_j - R'(b'_j)\right\|_2^2 + \left\|\overline{a_j} - \overline{b_j}\right\|_2^2}, \quad \text{by Pythagoras' Theorem} \\
&\leq \sqrt{\left\|a'_j - b'_j\right\|_2^2 + 2\sqrt{2}\left\|a'_j - b'_j\right\|_2 \left\|a'_1 - b'_1\right\|_2 + 2 \cdot \left\|a'_1 - b'_1\right\|_2^2 + \left\|\overline{a_j} - \overline{b_j}\right\|_2^2} \\
&\leq \sqrt{\|a_j - b_j\|_2^2 + 2\sqrt{2}\left\|a'_j - b'_j\right\|_2 \left\|a'_1 - b'_1\right\|_2 + 2 \cdot \left\|a'_1 - b'_1\right\|_2^2}, \text{ by Pythagoras' Theorem} \\
&\leq \sqrt{\|a_j - b_j\|_2^2 + 2\sqrt{2}\|a_j - b_j\|_2 \|a_1 - b_1\|_2 + 2 \cdot \|a_1 - b_1\|_2^2}, \text{ since } \left\|a'_j - b'_j\right\|_2 \leq \|a_j - b_j\|_2 \\
&\leq \|a_j - b_j\|_2 + \sqrt{2}\,\|a_1 - b_1\|_2 \\
&\leq (1 + \sqrt{2}) \cdot D^2_{\mathcal{B}}(A, B).
\end{aligned}$$

2. Let $\phi' \in (\pi/2, \pi]$. For every matched pair, i.e., for every $j = 1, \ldots, n$ we have

$$\begin{aligned}
\|a_j - R'(b_j)\|_2 &= \left\|a'_j + \overline{a_j} - R'(b'_j) - \overline{b_j}\right\|_2 \\
&= \sqrt{\left\|a'_j - R'(b'_j)\right\|_2^2 + \left\|\overline{a_j} - \overline{b_j}\right\|_2^2}, \quad \text{by Pythagoras' Theorem} \\
&\leq \sqrt{\left(\left\|a'_j - p^*\right\|_2 + \left\|p^* - R'(b'_j)\right\|_2\right)^2 + \left\|\overline{a_j} - \overline{b_j}\right\|_2^2} \\
&= \sqrt{\left(\left\|a'_j - p^*\right\|_2 + \left\|p^* - b'_j\right\|_2\right)^2 + \left\|\overline{a_j} - \overline{b_j}\right\|_2^2} \\
&\leq \sqrt{\left(\|b'_1 - p^*\|_2 + \|p^* - b'_1\|_2\right)^2 + \left\|\overline{a_j} - \overline{b_j}\right\|_2^2} \\
&\leq \sqrt{4\,\|b'_1 - a'_1\|_2^2 + \left\|\overline{a_j} - \overline{b_j}\right\|_2^2}, \quad \text{since } \phi' \in (\pi/2, \pi] \\
&\leq \sqrt{4\,\|b_1 - a_1\|_2^2 + \|a_j - b_j\|_2^2} \\
&\leq \sqrt{5} \cdot D_{\mathcal{B}}^2(A, B) \\
&< (1 + \sqrt{2}) \cdot D_{\mathcal{B}}^2(A, B).
\end{aligned}$$

Using these results:

$$\begin{aligned}
D_{\mathcal{B}}^2(A, R'(B)) &= \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left\|a_j - R'(b_{\pi(j)})\right\|_2 \\
&\leq \max_{j=1,\ldots,n} \|a_j - R'(b_j)\|_2\,, \quad \text{choosing the identity as permutation} \\
&\leq (1 + \sqrt{2}\,) \cdot D_{\mathcal{B}}^2(A, B).
\end{aligned}$$

□

Note that the approximation factor of $1+\sqrt{2}$ in the last lemma can be improved by choosing an angle smaller than $\pi/2$ for the case distinction. However, the approximation factor is greater than 2, even in the 2-dimensional case.

In the last proof we have made a case distinction and considered the case where $\phi' > \pi/2$. This case, at first glance, seems to be impossible. That this case can actually arise is shown in the following example:

Example. Two point sets in the plane showing the existence of such a case are illustrated in Figure 6.8. As one can easily check, the constellation given describes the minimum bottleneck distance under rotations around p^*, since at least one of the points $b_2, \ldots, b_5$ has to be matched to at least one of the points a_2, a_3 and another point of $b_2, \ldots, b_5$ has to be matched to one of the points a_4 and a_5. Therefore, any rotation of B would increase one of the distances between these two pairs. The angles given by b_1, p^* and any point of A are strictly larger than $\pi/2$.

We generalize the last result to every L_p-distance, where $1 \leq p \leq \infty$. The proof is analogous to the proof of Lemma 3.5 and is omitted.

Lemma 6.7. *Let $A, B \in \mathcal{P}^{d,n}$ be two point sets in dimension $d \geq 2$ and let S^* be a fixed d'-dimensional affine space with $0 \leq d' < d - 1$. Let $A \setminus S^* \neq \emptyset$ and $B \setminus S^* \neq \emptyset$. Let $\mathrm{Rot}(S^*)$ be the set of rotations leaving S^* invariant. Let $b_1 \in B$ be a point among all points of both sets with maximum Euclidean distance to S^*. Let $1 \leq p \leq \infty$. There exists a rotation $R' \in \mathrm{Rot}(S^*)$ such that*

$$D_{\mathcal{B}}^p(A, R'(B)) \;\leq\; (1 + \sqrt{2}\,)\sqrt{d} \cdot \min_{R \in \mathrm{Rot}(S^*)} D_{\mathcal{B}}^p(A, R(B)),$$

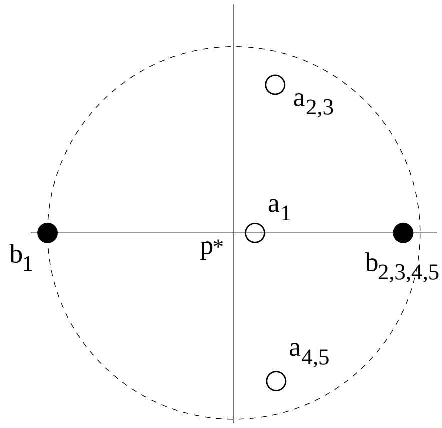


Figure 6.8: An angle $\geq \pi/2$ is possible.

where R' rotates B such that S^*, $R'(b_1)$ and some point of $A \setminus S^*$ are in a $(d'+1)$-dimensional space.

Note that in the above lemma we are using the furthest point to the rotation center with respect to the Euclidean distance to determine a rotation fulfilling the claim. This is important since the furthest point with respect to other distance measures on the ground set may vary while rotating the set.

Approximation for Rigid Motions in the Plane - An Improved Version

In this section we show how to use Algorithm 6.4 and Lemmas 6.6 and 6.7 to construct a fast approximation algorithm for rigid motions in the plane. We address higher dimensions in the following section.

Algorithm 6.5.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.
2. Let $b'_1 \in B'$ be a point with maximum Euclidean distance to the rotation center $r(A)$.
3. Find an optimal matching of A and B' under rotations of B' around $r(A)$, where $r(A), b'_1$ and a point of A are aligned.
   Let B'' be the image of B' under this rotation.
4. Output B'' together with the distance $D^p_{\mathcal{B}}(A, B'')$.

Theorem 6.18. *Let $A, B \in \mathcal{P}^{2,n}$ be two planar point sets. Let r be a $D^2_{\mathcal{B}}$-reference point with respect to rigid motions and with Lipschitz constant c. Algorithm 6.5 finds an approximately optimal matching for rigid motions with approximation factor $(1+\sqrt{2})(c+1)$ in time $O(T^{\text{ref}}(n) + n^{2.5}\log n)$.*

Proof. Using Lemma 6.6 we obtain a $(1+\sqrt{2})$-approximation for the bottleneck distance under rotations of B' around $r(A)$ by computing the minimum bottleneck distance among all rotations aligning b_1', $r(A)$ and some point of A. Thus, the total number of computations in the plane is $O(n)$, where each needs $O(n^{1.5}\log n)$ time. The claim follows by our abstract approximation Algorithm 1.2, see Theorem 1.7. $\square$

The latter algorithm decreases the runtime by a factor of $O(n)$ in comparison to the one described in Section 6.10.2. Unfortunately, as already introduced in the beginning of this section, we have to pay for the speed up by a slightly increased approximation ratio.

We generalize the result to any L_p-distance, where $1 \leq p \leq \infty$. The proof is analogous to the proof of Theorem 6.18, using Lemma 6.7 instead of Lemma 6.6.

Theorem 6.19. *Let $A, B \in \mathcal{P}^{2,n}$ be planar point sets and let $1 \leq p \leq \infty$. Let r be a $D_{\mathcal{B}}^p$-reference point with respect to rigid motions and with Lipschitz constant c. Algorithm 6.5 finds an approximately optimal matching for rigid motions with approximation factor $(2+\sqrt{2})(c+1)$ in time $O(T^{\text{ref}}(n) + n \cdot T^{D_{\mathcal{B}}^p})$.*

We apply the center of mass as a $D_{\mathcal{B}}^p$-reference point to the last two theorems:

Corollary 6.4. *For point sets in the plane, Algorithm 6.5 using the center of mass as a $D_{\mathcal{B}}^p$-reference point induces an approximation algorithm with approximation factor $2(1+\sqrt{2})$ in the Euclidean case and $2(2+\sqrt{2})$ for arbitrary L_p-distance, where $1 \leq p \leq \infty$. Its runtime is $O(n \cdot T^{D_{\mathcal{B}}^p}(n))$.*

Approximation Algorithms for Rigid Motions in Higher Dimensions - An Improved Version.

We use the following corollary of Lemma 6.6 to obtain an improved approximation algorithm for the bottleneck distance under rigid motions in arbitrary dimension $d \geq 3$. The proof is similar to the proof of Corollary 3.5, except for the fact that in this case we align the furthest point. Recall that for affine spaces $S_1, \ldots, S_n$ we use $\text{aff}(S_1, \ldots, S_n)$ to denote the affine space spanned by $S_1, \ldots, S_n$.

Corollary 6.5. *Let $A, B \in \mathcal{P}^{d,n}$ be two point sets in dimension $d \geq 3$. Let A, B be full-dimensional, that is $\dim A = \dim B = d$. Let $p^* \in \mathbb{R}^d$ be an arbitrary point. There are sequences of points $a^{(1)}, \ldots, a^{(d-1)} \in A$ and $b^{(1)}, \ldots, b^{(d-1)} \in B$, and a sequence of rotations $R^{(1)}, \ldots, R^{(d-1)} \in \text{Rot}(p^*)$, such that the following conditions hold for any $1 \leq k \leq d-1$:*

1. *$S^{(0)} = p^*$ and for $k > 0$*
$$S^{(k)} = \text{aff}\left(p^*, a^{(1)}, \ldots, a^{(k)}, R^{(k)} \circ \cdots \circ R^{(1)}(b^{(1)}), \ldots, R^{(k)} \circ \cdots \circ R^{(1)}(b^{(k)})\right),$$
2. *$R^{(k)}$ leaves $S^{(k-1)}$ fix,*
3. *$R^{(k)}$ aligns $a^{(k)}$ and $b^{(k)}$ with respect to $S^{(k-1)}$,*
4. *$a^{(k)}$ or $b^{(k)}$ is furthest to $S^{(k-1)}$,*
5. *$D_{\mathcal{B}}^2(A, R^{(k)} \circ \cdots \circ R^{(1)}(B)) \leq (1+\sqrt{2})^k \cdot \min_{R \in \text{Rot}(p^*)} D_{\mathcal{B}}^2(A, R(B))$.*

Proof. We prove the result by induction on k. For $k = 1$ we can apply Lemma 6.6. Let $1 \leq k < d-1$. By induction we have $\dim S^{(k)} = k$ and

$$D^2_{\mathcal{B}}(A, R^{(k)} \circ \cdots \circ R^{(1)}(B)) \;\leq\; (1+\sqrt{2}\,)^k \cdot \min_{R \in \mathrm{Rot}(p^*)} D^2_{\mathcal{B}}(A, R(B)).$$

Let $R^* := \arg\min_{R \in \mathrm{Rot}(S^{(k)})} D^2_{\mathcal{B}}(A, R \circ R^{(k)} \circ \cdots \circ R^{(1)}(B))$. Of course,

$$\begin{aligned} D^2_{\mathcal{B}}(A, R^* \circ R^{(k)} \circ \cdots \circ R^{(1)}(B)) &\leq D^2_{\mathcal{B}}(A, R^{(k)} \circ \cdots \circ R^{(1)}(B)) \\ &\leq (1+\sqrt{2}\,)^k \cdot \min_{R \in \mathrm{Rot}(p^*)} D^2_{\mathcal{B}}(A, R(B)). \end{aligned}$$

W.l.o.g. let $R^{(k)} \circ \cdots \circ R^{(1)}(b^{(k+1)})$ be the furthest point to $S^{(k)}$ among all points in $(A \cup R^{(k)} \circ \cdots \circ R^{(1)}(B)) \setminus S^{(k)}$. Since A and B are full-dimensional this point exists and we can apply Lemma 6.6 to find a point $a^{(k+1)}$ and a rotation $R^{(k+1)} \in \mathrm{Rot}(S^{(k)})$, such that $a^{(k+1)}$ and $R^{(k+1)} \circ \cdots \circ R^{(1)}(b^{(k+1)})$ are aligned with respect to $S^{(k)}$ and

$$\begin{aligned} D^2_{\mathcal{B}}(A, R^{(k+1)} \circ \cdots \circ R^{(1)}(B)) &\leq (1+\sqrt{2}\,) \cdot D^2_{\mathcal{B}}(A, R^* \circ R^{(k)} \circ \cdots \circ R^{(1)}(B)) \\ &\leq (1+\sqrt{2}\,)^{k+1} \cdot \min_{R \in \mathrm{Rot}(p^*)} D^2_{\mathcal{B}}(A, R(B)). \end{aligned}$$

□

Based on the abstract Algorithm 1.2 in Section 1.5.3 we get an approximation algorithm for minimizing the Euclidean bottleneck distance under rigid motions:

Theorem 6.20. *Let $A, B \in \mathcal{P}^{d,n}$, where $d \geq 3$. Let A, B be full-dimensional, that is $\dim A = \dim B = d$. Let $r\colon \mathcal{P}^{d,n} \to \mathbb{R}^d$ be a $D^2_{\mathcal{B}}$-reference point with respect to rigid motions and with Lipschitz constant c. An approximate rigid motion M' such that*

$$D^2_{\mathcal{B}}(A, M'(B)) \;\leq\; (1+\sqrt{2}\,)^{d-1}(c+1) \cdot D^{2\,\mathrm{opt}}_{\mathcal{B}}(A, B)$$

can be found in $O(T^{\mathrm{ref}}(n) + n^{d-1} \cdot T^{D^2_{\mathcal{B}}}(n))$ time.

Proof. As in the planar case we first translate B such that the reference points of A and B coincide. Algorithmically we proceed in the following way: We determine a furthest point to the rotation center p^*. W.l.o.g. let $b^{(1)}$ be this point. For every point $a^{(1)} \in A$ we do the following: We rotate B around the reference point such that $a^{(1)}, b^{(1)}$ and the reference point are aligned. We fix the line determined by these points. We determine a furthest point to this line, w.l.o.g. let $b^{(2)}$ be this point. For every point $a^{(2)} \in A$ we do the following: We rotate B around the fixed line such that $b^{(2)}$ and $a^{(2)}$ are aligned with respect to the fixed line. We fix the plane, determine a furthest point to this plane and continue until a $(d-1)$-dimensional subspace is fixed. We compute the bottleneck distance between A and the rotated version of B and continue.

Altogether we compute the bottleneck distance for every sequence of pairs of points as described in Corollary 6.5. Thus we have surely considered those sequences $a^{(1)}, \ldots, a^{(d-1)}$ and $b^{(1)}, \ldots, b^{(d-1)}$ and the corresponding rotation $R := R^{(d-1)} \circ \cdots \circ R^{(1)}$ that induce the approximation property in Corollary 6.5. Since by Lemma 3.7 the rotation R is unique, it equals the rotation constructed above and therefore we have proven the approximation property of the theorem.

Observing that there are $O(n^{d-1})$ possibilities for these sequences proves the runtime of the algorithm. For the runtime we further assume that the time to find a furthest point to a subspace is dominated by the time to compute the bottleneck distance. □

Note that if $d' := \dim B < \dim A$ it is enough to stop after d' steps. After this, every rotation leaves B invariant and thus does not change the bottleneck distance between the two sets. If $\dim A < \dim B$, we exchange the roles of A and B.

Similar to the proof of Lemma 3.5 we can extend the result to the bottleneck distance defined on any L_p-norm:

Theorem 6.21. *Let $A, B \in \mathcal{P}^{d,n}$, where $d \geq 3$. Let A, B be full-dimensional, that is $\dim A = \dim B = d$. Let $1 \leq p \leq \infty$. Let $r\colon \mathcal{P}^{d,n} \to \mathbb{R}^d$ be a $D_{\mathcal{B}}^p$-reference point with respect to rigid motions and with Lipschitz constant c. An approximate rigid motion M' such that*

$$D_{\mathcal{B}}^p(A, M'(B)) \ \leq\ (1+\sqrt{2}\,)^{d-1}\sqrt{d}(c+1)\cdot D_{\mathcal{B}}^{p\,\mathrm{opt}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(n) + n^{d-1}\cdot T^{D_{\mathcal{B}}^p}(n))$ time.

We apply the center of mass and obtain the following corollary:

Corollary 6.6. *Applying the center of mass as a $D_{\mathcal{B}}^p$-reference point to the algorithm described above, we obtain an approximation algorithm with approximation factor $2(1+\sqrt{2}\,)^{d-1}$ in the case of the Euclidean norm and $2(1+\sqrt{2}\,)^{d-1}\sqrt{d}$ for any other L_p-norm, where $1 \leq p \leq \infty$. The runtime is $O(n^{d-1}\cdot T^{D_{\mathcal{B}}^p}(n))$.*

Using δ-Nets for a Better Approximation Ratio

In this section we show how to use δ-nets, see Gonzalez [29], Clarkson [18] and the proof of Theorem 5.7, to improve the approximation ratio given in Lemma 6.6.

Lemma 6.8. *Let $A, B \in \mathcal{P}^{d,n}$ be two point sets and let $S^* \subset \mathbb{R}^d$ be a fixed d'-dimensional affine space with $0 \leq d' < d-1$. Let $0 < \varepsilon < 1/16$. Let $A \setminus S^* \neq \emptyset$ and $B \setminus S^* \neq \emptyset$. Let $\mathrm{Rot}(S^*)$ be the set of rotations leaving S^* invariant. Let $b_1 \in B$ be a point among all points of both sets with maximum distance to the rotation center S^*. There exists a rotation $R^{\mathrm{apx}} \in \mathrm{Rot}(S^*)$ such that*

$$D_{\mathcal{B}}^2(A, R^{\mathrm{apx}}(B)) \ \leq\ (2+\varepsilon)\cdot \min_{R\in\mathrm{Rot}(S^*)} D_{\mathcal{B}}^2(A, R(B)).$$

We can find this rotation by $O\left(n + \varepsilon^{-1/2}\right)$ computations of the bottleneck distance in $\mathbb{R}^d$.

Proof. Basically we use the same proof as for Lemma 6.6. Observing the analysis for case 1 of the case distinction we see that the approximation is bad if $\sin(\phi'/2)/\sin(\phi')$ gets close to $1/\sqrt{2}$, that is, for angles ϕ' close to $\pi/2$. Therefore we bound the angle ϕ' by computing the bottleneck distance for every rotation aligning b_1' and at least one of the points a_j', and for every rotation on a sufficiently fine uniformly distributed grid on the unit circle. By choosing the grid fine enough we additionally have that $\phi' \leq \pi/2$ and thus the second case of the case distinction does not occur.

Let $\delta := 1/2\cdot\sqrt{1-1/(1+\varepsilon)^2}$. Let Y be a δ-net on the boundary of the unit circle $\mathbb{S}^1$. Let R^{apx} be a rotation inducing the minimum $D_{\mathcal{B}}^2(A, R_\beta(B))$, minimized over all angles β corresponding to points in Y and rotations aligning b_1' and at least one of the points a_j'. Let R' be the closest rotation to the optimal rotation R^{opt}. Let α be the angle between R' and R^{opt}.

Now, we have

$$\begin{aligned} 2\,\frac{\sin(\alpha/2)}{\sin(\alpha)} &= 2\,(2\cdot\cos(\alpha/2))^{-1} \\ &= \left(\sqrt{1-\sin^2(\alpha/2)}\right)^{-1} \\ &\le \left(\sqrt{1-(\delta/2)^2}\right)^{-1} \\ &= \left(\sqrt{1-(1-1/(1+\varepsilon)^2)}\right)^{-1} \\ &= 1+\varepsilon. \end{aligned}$$

Therefore, analogously to the proof of Lemma 6.6 for every matched pair, i.e., for every $j = 1, \ldots, n$ we have

$$\left\|a'_j - R'(b'_j)\right\|_2 \;\le\; \left\|a'_j - b'_j\right\|_2 + (1+\varepsilon)\,\|a'_1 - b'_1\|_2$$

and further

$$\|a_j - R'(b_j)\|_2 \;\le\; (2+\varepsilon)\cdot D_{\mathcal{B}}(A,B).$$

Similarly to the proof of Lemma 6.6 we have

$$D^2_{\mathcal{B}}(A, R^{\mathrm{apx}}(B)) \;\le\; D^2_{\mathcal{B}}(A, R'(B)) \;\le\; (2+\varepsilon)\cdot D^2_{\mathcal{B}}(A,B).$$

Since

$$\delta \;=\; 1/2\cdot\sqrt{1-\frac{1}{(1+\varepsilon)^2}} \;=\; 1/2\cdot\frac{\sqrt{(2+\varepsilon)\,\varepsilon}}{1+\varepsilon} = O\left(\varepsilon^{1/2}\right)$$

the number of grid points is $O\left(\delta^{-1}\right) \;=\; O\left(\varepsilon^{-1/2}\right)$. □

The last lemma describes an approximation algorithm in the planar case, thus it directly proves the following theorem. It is open if it can be generalized to arbitrary dimension $d > 2$.

Theorem 6.22. *Let $A, B \in \mathcal{P}^{2,n}$ be planar point sets. Let $0 < \varepsilon < 1/16$. Let $r\colon \mathcal{P}^{2,n} \to \mathbb{R}^2$ be a $D^2_{\mathcal{B}}$-reference point with respect to rigid motions and with Lipschitz constant c. An approximate rigid motion M' such that*

$$D^2_{\mathcal{B}}(A, M'(B)) \;\le\; (2+\varepsilon)(c+1)\cdot D^{2\,\mathrm{opt}}_{\mathcal{B}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(n) + (n+\varepsilon^{-1/2})\cdot n^{1.5}\log n)$ time.

The proof of the generalization to arbitrary L_p-distance is analogous to the proof of Lemma 3.5 and omitted.

Theorem 6.23. *Let $A, B \in \mathcal{P}^{2,n}$ be planar point sets. Let $0 < \varepsilon < 1/16$ and let $1 \le p \le \infty$. Let $r\colon \mathcal{P}^{2,n} \to \mathbb{R}^2$ be a $D^2_{\mathcal{B}}$-reference point with respect to rigid motions and with Lipschitz constant c. An approximate rigid motion M' such that*

$$D^p_{\mathcal{B}}(A, M'(B)) \;\le\; (2+\varepsilon)(c+1)\sqrt{2}\cdot D^{p\,\mathrm{opt}}_{\mathcal{B}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(n) + (n+\varepsilon^{-1/2})\cdot T^{D^p_{\mathcal{B}}}(n))$ time.

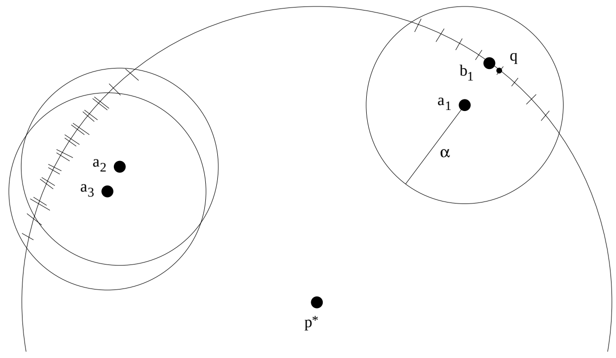


Figure 6.9: $(1+\varepsilon)$-Approximation for Rotations

6.10.4 FPTAS for Rotations in the Plane

We use the constant-factor approximation derived in Section 6.10.3 to obtain a $(1+\varepsilon)$-approximation for the bottleneck distance under rotations around a fixed point in the plane.

As we have observed in Section 6.10.3 it suffices to take care that the furthest point lies in the neighborhood of the point matched to it in a perfect matching inducing the bottleneck distance. We do this by computing the bottleneck distance on a sufficiently fine grid in rotation space around the rotation aligning those two points. Since we do not know this rotation, we place the grid around every rotation aligning the furthest point and a point of the other set. Thus we have to compute the bottleneck distance at $O(\varepsilon^{-1})$ angles for each of the n points of the other set.

Theorem 6.24. *Let $0<\varepsilon<1$. Let $A, B \in \mathcal{P}^{2,n}$ be two planar point sets and let $p^* \in \mathbb{R}^2$ be a fixed point. Let $\mathrm{Rot}(p^*)$ be the set of rotations around p^*. We can find a rotation $R^\varepsilon \in \mathrm{Rot}(p^*)$ such that*

$$D^2_{\mathcal{B}}(A, R^\varepsilon(B)) \;\le\; (1+\varepsilon)\cdot \min_{R\in\mathrm{Rot}(p^*)} D^2_{\mathcal{B}}(A, R(B))$$

in $O(\varepsilon^{-1}n^{2.5}\log n)$ time.

Proof. W.l.o.g. let the point sets A, B be in optimal position with respect to rotations of B around p^*. We determine a value α, such that $D^2_{\mathcal{B}}(A,B) \le \alpha \le (1+\sqrt{2}\,)\cdot D^2_{\mathcal{B}}(A,B)$ using the approximation derived in Lemma 6.6. For the following construction see Figure 6.9. Let $b_1 \in B$ be a point among all points of both sets with maximum Euclidean distance to the rotation center p^*. Let K be the circle through b_1 around the rotation center p^*. For any $1 \le j \le n$, let $\Delta_\alpha(a_j)$ denote the disc around a_j with radius α. Let d_j be the circular arc defined by $d_j := \Delta_\alpha(a_j)\cap K$. Since $D^2_{\mathcal{B}}(A,B)\le\alpha$, b_1 has to lie in one of the discs $\Delta_\alpha(a_j)$ and therefore has to be located on d_j. The length $|d_j|$ of d_j is bounded by the perimeter of the circle around a_j, that is, $|d_j| \le 2\pi\alpha$. Now, we subdivide d_j into pieces of length $\varepsilon\alpha/(1+\sqrt{2}\,)$. The number of pieces is at most $2(1+\sqrt{2}\,)\pi\alpha/(\varepsilon\alpha) = O(\varepsilon^{-1})$. Let $\mathcal{A}_j$ denote the set of angles aligning b_1 and the subdivision points on d_j. W.l.o.g. let b_1 be matched to a_1, that is, b_1 is located on d_1. Let $q \in d_1$ be the subdivision point closest to b_1. Let $\phi_q \in \mathcal{A}_1$ be an angle, such that the rotation R_{ϕ_q} rotates B in a way that b_1 and q coincide. Since b_1 and q are on d_1, the distance between these two points is at most $\varepsilon\alpha/(1+\sqrt{2}\,)$. Now,

$$\begin{aligned}
D_{\mathcal{B}}^2(A, R_{\phi_q}(B)) &\le \max_{j=1,\dots,n} \left\|a_j - R_{\phi_q}(b_j)\right\|_2 \\
&\le \max_{j=1,\dots,n} \|a_j - b_j\|_2 + \max_{j=1,\dots,n} \left\|b_j - R_{\phi_q}(b_j)\right\|_2 \\
&\le D_{\mathcal{B}}^2(A,B) + \left\|b_1 - R_{\phi_q}(b_1)\right\|_2, \qquad \text{since } b_1 \text{ is furthest} \\
&= D_{\mathcal{B}}^2(A,B) + \|b_1 - q\|_2 \\
&\le D_{\mathcal{B}}^2(A,B) + \frac{\varepsilon\alpha}{(1+\sqrt{2}\,)} \\
&\le (1+\varepsilon)\cdot D_{\mathcal{B}}^2(A,B).
\end{aligned}$$

Computing the bottleneck distance at all $O(n\varepsilon^{-1})$ subdivision points in time $O(n^{1.5}\log n)$ leads to the approximation algorithm and completes the proof. □

We have proven the last lemma for the Euclidean case. The ideas involved also hold for arbitrary L_p-distance, where $1 \le p \le \infty$. The constants involved have to be adjusted.

6.10.5 FPTAS for Rigid Motions in the Plane

It is straightforward to construct an FPTAS for rigid motions in the plane. The method is similar to the one used by Giannopoulos [26] to construct an approximation algorithm for the EMD_2.

Algorithm 6.6.

1. Run Algorithm 6.5 to determine α, such that

$$D_{\mathcal{B}}^{2\,\mathrm{opt}}(A, M(B)) \;\le\; \alpha \;\le\; (1+\sqrt{2}\,)(c+1)\cdot D_{\mathcal{B}}^{2\,\mathrm{opt}}(A, M(B)).$$

2. Let $\tau^{\mathrm{ref}} := r(A) - r(B)$ be the approximate translation.

3. Place an axis-parallel cube C of side length $2c\alpha$ centered at τ^{ref}.

4. Consider a grid Γ centered at the origin with cell size $\gamma := \varepsilon\alpha/((2\sqrt{2}+4)(c+1))$.

5. For every grid point g inside C do:
   Compute a rotation $R^\varepsilon \in \mathrm{Rot}(g)$, such that

$$D_{\mathcal{B}}^2(A, R^\varepsilon(B+g)) \;\le\; (1+\varepsilon/3)\cdot \min_{R\in\mathrm{Rot}(g)} D_{\mathcal{B}}^2(A, R(B+g))$$

   according to Theorem 6.24.

6. Output the minimum of all rotations computed in 5.

Theorem 6.25. *Let $0 < \varepsilon < 1$. Let $A, B \in \mathcal{P}^{2,n}$ be two planar point sets. Let r be a $D_{\mathcal{B}}^p$-reference point with respect to rigid motions and with Lipschitz constant c. Algorithm 6.6 finds a rigid motion $M^\varepsilon \in \mathrm{Rot}(p^*)$ such that*

$$D_{\mathcal{B}}^2(A, M^\varepsilon(B)) \;\le\; (1+\varepsilon)\cdot D_{\mathcal{B}}^{2\,\mathrm{opt}}(A, M(B))$$

in $O(\varepsilon^{-3}n^{2.5}\log n)$ time.

Proof. Let $0 < \varepsilon < 1$. Let $M^{\mathrm{opt}} := R^{\mathrm{opt}} \circ \tau^{\mathrm{opt}}$ be an optimal rigid motion of B, where $\tau^{\mathrm{opt}} \in \mathbb{R}^2$ is a translation and R^{opt} is a rotation around $\tau^{\mathrm{opt}}(r(B))$. We show that τ^{opt} is inside the cube C:

$$\begin{aligned}
\left\|\tau^{\mathrm{ref}} - \tau^{\mathrm{opt}}\right\|_2 &= \left\|r(A) - r(B) - \tau^{\mathrm{opt}}\right\|_2 \\
&= \left\|r(A) - \tau^{\mathrm{opt}}(r(B))\right\|_2 \\
&= \left\|r(A) - R^{\mathrm{opt}} \circ \tau^{\mathrm{opt}}(r(B))\right\|_2, \quad \text{since } R^{\mathrm{opt}} \text{ rotates around } \tau^{\mathrm{opt}}(r(B)) \\
&= \left\|r(A) - r(R^{\mathrm{opt}} \circ \tau^{\mathrm{opt}}(B))\right\|_2, \quad \text{by the equivariance of } r \\
&\leq c \cdot D^2_{\mathcal{B}}(A, M^{\mathrm{opt}}(B)) \\
&\leq c\alpha.
\end{aligned}$$

Let g be the grid point closest to τ^{opt}. Let $R^{\varepsilon,g}$ be a rotation around g, such that

$$D^2_{\mathcal{B}}(A, R^{\varepsilon,g} \circ g(B)) \;\leq\; (1 + \varepsilon/3) \cdot D^2_{\mathcal{B}}(A, R^{\mathrm{opt},g} \circ g(B)),$$

where $R^{\mathrm{opt},g}$ denotes the optimal rotation of B around g. By Theorem 6.24 we can compute $R^{\varepsilon,g}$ in $O(\varepsilon^{-1} n^{2.5} \log n)$ time for a fixed translation g. Let M^{alg} be the rigid motion we get by Algorithm 6.6. Obviously, $D^2_{\mathcal{B}}(A, M^{\mathrm{opt}}(B)) \leq D^2_{\mathcal{B}}(A, M^{\mathrm{alg}}(B))$. Further,

$$\begin{aligned}
D^2_{\mathcal{B}}(A, M^{\mathrm{alg}}(B)) &\leq D^2_{\mathcal{B}}(A, R^{\varepsilon,g} \circ g(B)), \quad \text{since } g \text{ is a fixed grid point} \\
&\leq (1 + \varepsilon/3) \cdot D^2_{\mathcal{B}}(A, R^{\mathrm{opt},g} \circ g(B)) \\
&\leq (1 + \varepsilon/3) \cdot D^2_{\mathcal{B}}(A, R^{\mathrm{opt}} \circ g(B)) \\
&\leq (1 + \varepsilon/3) \left(D^2_{\mathcal{B}}(A, R^{\mathrm{opt}} \circ \tau^{\mathrm{opt}}(B)) + D^2_{\mathcal{B}}(R^{\mathrm{opt}} \circ \tau^{\mathrm{opt}}(B), D^2_{\mathcal{B}}(A, R^{\mathrm{opt}} \circ g(B)))\right) \\
&\leq (1 + \varepsilon/3) \left(D^2_{\mathcal{B}}(A, M^{\mathrm{opt}}(B)) + \|g - \tau^{\mathrm{opt}}\|_2\right) \\
&\leq (1 + \varepsilon/3) \left(D^2_{\mathcal{B}}(A, M^{\mathrm{opt}}(B)) + \sqrt{2}\, \frac{\varepsilon\alpha}{(2\sqrt{2} + 4)(c+1)}\right) \\
&\leq (1 + \varepsilon/3) \left(D^2_{\mathcal{B}}(A, M^{\mathrm{opt}}(B)) + \sqrt{2}\, \frac{\varepsilon(1 + \sqrt{2})(c+1) \cdot D^{2\,\mathrm{opt}}_{\mathcal{B}}(A, M(B))}{(2\sqrt{2} + 4)(c+1)}\right) \\
&= (1 + \varepsilon/3)(1 + \varepsilon/2) \cdot D^2_{\mathcal{B}}(A, M^{\mathrm{opt}}(B)) \\
&\leq (1 + \varepsilon) \cdot D^2_{\mathcal{B}}(A, M^{\mathrm{opt}}(B)).
\end{aligned}$$

The runtime of Algorithm 6.6 is the time to compute the $(1 + \varepsilon)$-approximation for the bottleneck distance under rotations in $O(\varepsilon^{-1} n^{2.5} \log n)$ time at the $O(\varepsilon^{-2})$ grid points. □

6.11 Similarities

In this section we present approximation algorithms for matching point sets under positive similarities with respect to the Euclidean bottleneck distance. More precisely, we want to compute $\min_S D^2_{\mathcal{B}}(A, S(B))$, where the minimum is taken over all positive similarities S. Again, negative similarities can be handled by running the algorithms twice, once with a reflected version of one of the shapes. See also the remarks in Sections 1.5.4 and 3.4.5. The approach is to use the approximation algorithm for rigid motions where B is scaled by $d^{(2)}_{r(A)}(A)/d^{(2)}_{r(B)}(B)$. In general, for a fixed point $p^* \in \mathbb{R}^d$ and a point set $A \in \mathcal{P}^{d,n}$, $d^{(p)}_{p^*}(A)$ is defined as the distance of a furthest point in A to p^*, i.e., $d^{(p)}_{p^*}(A) = \max_{a \in A} \|a - p^*\|_p$. We will only use the furthest point with respect to the Euclidean distance to obtain an approximation on the scaling ratio. This is important since the furthest point with respect to arbitrary L_p-distance is not invariant under rotations.

Algorithm 6.7.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.
2. Scale B' by $d^{(2)}_{r(A)}(A)/d^{(2)}_{r(B')}(B')$ around $r(A)$.
   Let B'' be the image of B' under this scaling.
3. Find an optimal matching of A and B'' under rotations of B'' around $r(A)$.
   Let B''' be the image of B'' under this rotation.
4. Output B''' together with the distance $D^p_{\mathcal{B}}(A, B''')$.

To show the correctness of this algorithm we use the following results. We prove the first lemma for arbitrary L_p-distance though we only need the Euclidean case later on.

Lemma 6.9. *Let $A \in \mathcal{P}^{d,n}$ be a point set. Let σ_1, σ_2 be scalings around the same center p^* with ratios γ_1 and γ_2, respectively. Then*

$$D^p_{\mathcal{B}}(\sigma_1(A), \sigma_2(A)) \;=\; |\gamma_1 - \gamma_2| \cdot d^{(p)}_{p^*}(A).$$

Proof.

$$\begin{aligned}
D^p_{\mathcal{B}}(\sigma_1(A), \sigma_2(A)) &= \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left\|\sigma_1(a_j) - \sigma_2(a_{\pi(j)})\right\|_p \\
&\leq \max_{j=1,\ldots,n} \|\sigma_1(a_j) - \sigma_2(a_j)\|_p \\
&= \|\sigma_1(a_{j^*}) - \sigma_2(a_{j^*})\|_p\,, \quad \text{for a fixed index } j^* \\
&= \|p^* + \gamma_1(a_{j^*} - p^*) - (p^* + \gamma_2(a_{j^*} - p^*))\|_p \\
&= \|(\gamma_1 - \gamma_2)(a_{j^*} - p^*)\|_p \\
&= |\gamma_1 - \gamma_2| \cdot \|a_{j^*} - p^*\|_p \\
&\leq |\gamma_1 - \gamma_2| \cdot d^{(p)}_{p^*}(A).
\end{aligned}$$

Let a be the furthest point in A to the center with respect to the L_p-distance, that is, $a = \arg\max_{j=1,\ldots,n} \|a_j - p^*\|_p$. W.l.o.g. let $\gamma_1 \geq \gamma_2$. Then

$$\begin{aligned}
&D_{\mathcal{B}}^p(\sigma_1(A), \sigma_2(A)) \\
\geq\ & \min_{j=1,\ldots,n} \|\sigma_1(a) - \sigma_2(a_j)\|_p \\
=\ & \min_{j=1,\ldots,n} \|\sigma_1(a) - p^* + p^* - \sigma_2(a_j)\|_p \\
\geq\ & \min_{j=1,\ldots,n} \left| \|\sigma_1(a) - p^*\|_p - \|p^* - \sigma_2(a_j)\|_p \right|, \quad \text{since } |\|x\|_p - \|y\|_p| \leq \|x+y\|_p \\
=\ & \min_{j=1,\ldots,n} \left(\|\sigma_1(a) - p^*\|_p - \|\sigma_2(a_j) - p^*\|_p \right), \quad \text{since } a \text{ is furthest and } \gamma_1 \geq \gamma_2 \\
=\ & \|\sigma_1(a) - p^*\|_p - \max_{j=1,\ldots,n} \|\sigma_2(a_j) - p^*\|_p \\
=\ & \|\sigma_1(a) - p^*\|_p - \|\sigma_2(a) - p^*\|_p, \quad \text{since } a \text{ is furthest} \\
=\ & \|\sigma_1(a) - \sigma_2(a)\|_p, \quad \sigma_1(a) \text{ and } \sigma_2(a) \text{ lie on a ray through } p^* \\
=\ & |\gamma_1 - \gamma_2| \cdot \|a - p^*\|_p, \quad \text{see the proof of "}\leq\text{"} \\
=\ & |\gamma_1 - \gamma_2| \cdot d_{p^*}^{(p)}(A).
\end{aligned}$$

□

We use the following lemma to prove a lower bound for the bottleneck distance.

Lemma 6.10. *Let $A, B \in \mathcal{P}^{d,n}$, and let p^* and q^* be two fixed points in $\mathbb{R}^d$. Then*

$$\left| d_{p^*}^{(p)}(A) - d_{q^*}^{(p)}(B) \right| \leq D_{\mathcal{B}}^p(A, B) + \|p^* - q^*\|_p .$$

Proof.

$$\begin{aligned}
&|d_{p^*}(A) - d_{q^*}(B)| \\
=\ & \left| \max_{j=1,\ldots,n} \|a_j - p^*\|_p - \max_{j=1,\ldots,n} \|b_j - q^*\|_p \right| \\
=\ & \min_{\pi \in S_n} \left| \max_{j=1,\ldots,n} \|a_j - p^*\|_p - \max_{j=1,\ldots,n} \left\|b_{\pi(j)} - q^*\right\|_p \right|, \quad \text{only a reordering} \\
\leq\ & \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left| \|a_j - p^*\|_p - \left\|b_{\pi(j)} - q^*\right\|_p \right|, \quad (1) \\
=\ & \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left| \|a_j - p^*\|_p - \left\|q^* - b_{\pi(j)}\right\|_p \right| \\
\leq\ & \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left\|a_j - p^* + q^* - b_{\pi(j)}\right\|_p, \quad \text{since } |\|x\|_p - \|y\|_p| \leq \|x+y\|_p \\
\leq\ & \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left(\left\|a_j - b_{\pi(j)}\right\|_p + \|q^* - p^*\|_p \right) \\
=\ & \|q^* - p^*\|_p + \min_{\pi \in S_n} \max_{j=1,\ldots,n} \left\|a_j - b_{\pi(j)}\right\|_p \\
=\ & \|q^* - p^*\|_p + D_{\mathcal{B}}^p(A, B)
\end{aligned}$$

To prove (1): Let $(x_j)_{j=1,\ldots,n}, (y_j)_{j=1,\ldots,n} \geq 0$ be two non-negative sequences of real numbers. W.l.o.g. let $\max_{j=1,\ldots,n} x_j \geq \max_{j=1,\ldots,n} y_j$. Then

$$\begin{aligned}\left|\max_{j=1,\ldots,n} x_j - \max_{j=1,\ldots,n} y_j\right| &= \max_{j=1,\ldots,n} x_j - \max_{j=1,\ldots,n} y_j \\ &= x_{j^*} - \max_{j=1,\ldots,n} y_j, \quad \text{for an index } j^* \in \{1,\ldots,n\} \\ &\leq x_{j^*} - y_{j^*}, \quad \text{since } 0 \leq y_{j^*} \leq \max\nolimits_{j=1,\ldots,n} y_j \\ &\leq |x_{j^*} - y_{j^*}| \\ &\leq \max_{j=1,\ldots,n} |x_j - y_j|\,.\end{aligned}$$

□

The following theorem gives a lower bound for the bottleneck distance based on the distance of the furthest points to their reference points.

Theorem 6.26. *Let $A, B \in \mathcal{P}^{d,n}$. Let r be a $D^p_{\mathcal{B}}$-reference point with respect to similarities and with Lipschitz constant c. Then*

$$\left|d^{(p)}_{r(A)}(A) - d^{(p)}_{r(B)}(B)\right| \;\leq\; (1+c)\cdot D^p_{\mathcal{B}}(A,B).$$

Proof. By Lemma 6.10 we have

$$\left|d^{(p)}_{r(A)}(A) - d^{(p)}_{r(B)}(B)\right| \;\leq\; \|r(A) - r(B)\|_p + D^p_{\mathcal{B}}(A,B).$$

The claim follows by the Lipschitz continuity. □

We state the following result proving that Algorithm 6.7 leads to a constant-factor approximation for the bottleneck distance under similarities in the Euclidean case. Using Lemma 6.9 and Theorem 6.26, the proof is analogous to the proof of the corresponding Theorem 3.18 for the EMD_2.

Theorem 6.27. *Let $A, B \in \mathcal{P}^{d,n}$. Let r be a $D^2_{\mathcal{B}}$-reference point with respect to similarities and with Lipschitz constant c. Algorithm 6.7 finds an approximately optimal matching for similarities with approximation factor $2(c+1)$ in time $O(T^{\mathrm{ref}}(n) + T^{D^2_{\mathcal{B}}}(n) + T^{\mathrm{rot}}(n))$.*

We generalize the last theorem to any L_p-distance, where $1 \leq p \leq \infty$. The proof is analogous to the proof of Theorem 3.5 and omitted.

Theorem 6.28. *Let $1 \leq p \leq \infty$. Let r be a $D^p_{\mathcal{B}}$-reference point with respect to similarities and with Lipschitz constant c. Algorithm 6.7 finds an approximately optimal matching for similarities with approximation factor $2\sqrt{d}\,(c+1)$ in time $O(T^{\mathrm{ref}}(n) + T^{D^p_{\mathcal{B}}}(n) + T^{\mathrm{rot}}(n))$.*

Applying the center of mass as a $D^p_{\mathcal{B}}$-reference point leads to the following corollary:

Corollary 6.7. *Algorithm 6.7 using the center of mass as a $D^p_{\mathcal{B}}$-reference point induces an approximation algorithm with approximation factor 4 in the Euclidean case and $4\sqrt{d}$ for arbitrary L_p-distance, where $1 \leq p \leq \infty$. The runtime is $O(T^{D^p_{\mathcal{B}}}(n) + T^{\mathrm{rot}}(n))$.*

Approximation for Similarities in the Plane

Analogous to Section 6.10.3 we get a fast approximation algorithm by combining Algorithm 6.7 and the method for approximating the optimal rotation presented in Lemma 6.6.

Algorithm 6.8.

1. Compute $r(A)$ and $r(B)$ and translate B by $r(A) - r(B)$.
   Let B' be the image of B.
2. Scale B' by $d^{(2)}_{r(A)}(A)/d^{(2)}_{r(B')}(B')$ around the center $r(A)$.
   Let B'' be the image of B' under this scaling.
3. Let $b''_1 \in B''$ be a point with maximum Euclidean distance to the rotation center $r(A)$.
4. Find an optimal matching of A and B'' under rotations of B'' around $r(A)$, where $r(A), b''_1$ and a point of A are aligned or b''_1 is matched to a coordinate axis of a coordinate system placed at $r(A)$.
   Let B''' be the image of B'' under this rotation.
5. Output B''' together with the distance $D^p_{\mathcal{B}}(A, B''')$.

The following theorem proves that Algorithm 6.8 leads to an efficient constant-factor approximation for the bottleneck distance under similarities in the Euclidean case. Again the proof is analogous to the proof of the corresponding Theorem 3.20 for the EMD_2.

Theorem 6.29. *Let $A, B \in \mathcal{P}^{2,n}$ be planar point sets. Let r be a $D^2_{\mathcal{B}}$-reference point with respect to similarities and with Lipschitz constant c. Algorithm 6.8 finds an approximately optimal matching for similarities with approximation factor $2(1+\sqrt{2})(c+1)$ in time $O(T^{\mathrm{ref}}(n)+n^{2.5}\log n)$.*

We generalize the last theorem to any L_p-distance, where $1 \le p \le \infty$. The proof is analogous to the proof of Theorem 3.5 and omitted.

Theorem 6.30. *Let $A, B \in \mathcal{P}^{2,n}$ be planar point sets. Let $1 \le p \le \infty$. Let r be a $D^p_{\mathcal{B}}$-reference point with respect to similarities and with Lipschitz constant c. Algorithm 6.7 finds an approximately optimal matching for similarities with approximation factor $2\sqrt{d}\,(1+\sqrt{2})(c+1)$ in time $O(T^{\mathrm{ref}}(n) + T^{D^p_{\mathcal{B}}}(n) + T^{\mathrm{rot}}(n))$.*

Applying the center of mass as a $D^p_{\mathcal{B}}$-reference point leads to the following corollary:

Corollary 6.8. *For planar point sets, Algorithm 6.7 using the center of mass as a $D^p_{\mathcal{B}}$-reference point induces an approximation algorithm with approximation factor $4(1+\sqrt{2})$ in the Euclidean case and $4\sqrt{d}(1+\sqrt{2})$ for arbitrary L_p-distance, where $1 \le p \le \infty$. The runtime is $O(T^{D^p_{\mathcal{B}}}(n)+T^{\mathrm{rot}}(n))$.*

Approximation Algorithms for Similarities in Higher Dimensions

We can apply the approach used in Section 6.10.3 to obtain an algorithm for similarities in dimensions ≥ 3. We can do this by applying the algorithm for rigid motions to A and B', where B' is B scaled by the quotient of the distances of the furthest points in each set. We just state the result. The proof is analogous to the proofs for rigid motions.

Theorem 6.31. *Let $A, B \in \mathcal{P}^{d,n}$, where $d \geq 3$. Let A, B be full-dimensional, that is* $\dim A = \dim B = d$. *Let $r\colon \mathcal{P}^{d,n} \to \mathbb{R}^d$ be a $D^2_{\mathcal{B}}$-reference point with respect to similarities and with Lipschitz constant c. An approximate similarity S' such that*

$$D^2_{\mathcal{B}}(A, S'(B)) \;\leq\; 2\,(1+\sqrt{2}\,)^{d-1}(c+1)\cdot {D^2_{\mathcal{B}}}^{\mathrm{opt}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(n) + n^{d-1}\cdot T^{D^2_{\mathcal{B}}}(n))$ time.

Similar to the proof of Lemma 3.5 we can extend the result to the bottleneck distance defined on any L_p-norm:

Theorem 6.32. *Let $A, B \in \mathcal{P}^{d,n}$, where $d \geq 3$. Let A, B be full-dimensional, that is* $\dim A = \dim B = d$. *Let $1 \leq p \leq \infty$. Let $r\colon \mathcal{P}^{d,n} \to \mathbb{R}^d$ be a $D^p_{\mathcal{B}}$-reference point with respect to similarities and with Lipschitz constant c. An approximate similarity S' such that*

$$D^p_{\mathcal{B}}(A, S'(B)) \;\leq\; 2\,(1+\sqrt{2}\,)^{d-1}\sqrt{d}(c+1)\cdot {D^p_{\mathcal{B}}}^{\mathrm{opt}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(n) + n^{d-1}\cdot T^{D^p_{\mathcal{B}}}(n))$ time.

Using δ-Nets for a Better Approximation Ratio

It is also possible to use the approach using δ-nets to get a better approximation for computing the bottleneck distance under similarities. This can be done by applying the algorithm for rigid motions, see Section 6.10.3, to A and B', where B' is B scaled by the quotient of the distances of the furthest points in each set. We just state the result and omit the proof which is analogous to the proof for rigid motions.

Theorem 6.33. *Let $A, B \in \mathcal{P}^{2,n}$ be planar point sets. Let $0 < \varepsilon < 1/16$. Let $r\colon \mathcal{P}^{2,n} \to \mathbb{R}^d$ be a $D^2_{\mathcal{B}}$-reference point with respect to similarities and with Lipschitz constant c. An approximate similarity S' such that*

$$D^2_{\mathcal{B}}(A, S'(B)) \;\leq\; 2\,(c+1)\,(2+\varepsilon)\cdot {D^2_{\mathcal{B}}}^{\mathrm{opt}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(n) + (n+\varepsilon^{-1/2})\cdot n^{1.5}\log n)$ time.

We can also prove the result for arbitrary L_p-distance, where $1 \leq p \leq \infty$:

Theorem 6.34. *Let $A, B \in \mathcal{P}^{2,n}$ be planar point sets. Let $0 < \varepsilon < 1/16$ and let $1 \leq p \leq \infty$. Let $r\colon \mathcal{P}^{2,n} \to \mathbb{R}^2$ be a $D^2_{\mathcal{B}}$-reference point with respect to similarities and with Lipschitz constant c. An approximate similarity S' such that*

$$D^p_{\mathcal{B}}(A, S'(B)) \;\leq\; 2\,(2+\varepsilon)(c+1)\sqrt{2}\cdot {D^p_{\mathcal{B}}}^{\mathrm{opt}}(A,B)$$

can be found in $O(T^{\mathrm{ref}}(n) + (n+\varepsilon^{-1/2})\cdot T^{D^p_{\mathcal{B}}}(n))$ time.

Chapter 7

Other Distance Measures

In this chapter we give a short overview about reference point methods or similar approaches for distance measures on shapes which can be found in the literature. We consider the Fréchet distance, the volume of the symmetric difference, the volume of overlap and the Frobenius norm and embed the result in our reference point framework.

7.1 Fréchet Distance

For several applications in the area of shape matching one wants to compute the similarity of parameterized curves. Efficiently computable distance measures are known for this problem if the parameterizations of the curves are fixed. However, it may be more interesting to compute the distance between the geometry of the curves and thus independent of the given parameterization. A distance measure fulfilling this demand is the Fréchet distance, see for example the work of Alt and Godau [7], and Godau [28]. We define a curve in $\mathbb{R}^d$ as a continuous function from the unit interval $[0,1]$ into $\mathbb{R}^d$. Let f, g be two curves and let $1 \leq p \leq \infty$. Let $\mathrm{Hom}([0,1])$ be the set of homeomorphisms from $[0,1]$ onto itself with the additional constraint that for all $\alpha \in \mathrm{Hom}([0,1])$ we have $\alpha(0) = 0$ and $\alpha(1) = 1$. Then

$$D_{\mathcal{F}}^p(f,g) \;:=\; \inf_{\alpha,\beta \in \mathrm{Hom}([0,1])} \max_{t \in [0,1]} \|f(\alpha(t)) - g(\beta(t))\|_p$$

denotes the Fréchet distance between f and g.

Alt, Knauer, and Wenk [8] prove the following result:

Theorem 7.1. *[8] The starting point $f(0)$ of a curve $f : [0,1] \to \mathbb{R}^d$ is a reference point for the Fréchet distance with respect to translations. Its Lipschitz constant is* 1.

We can extend this result easily by observing that the starting point of a curve is also equivariant under affine transformations. Since the Lipschitz continuity is independent of the considered class of transformations we have the following sharper result:

Theorem 7.2. *The starting point $f(0)$ of a curve $f : [0,1] \to \mathbb{R}^d$ is a reference point for the Fréchet distance with respect to affine transformations. Its Lipschitz constant is* 1.

It is easy to see that also the end point $f(1)$ is a reference point for the Fréchet distance with respect to affine transformations. Note that it is not true that $f(\alpha)$ is a reference point for any $\alpha \in [0,1]$. Instead the following result is a corollary from Theorem 1.2:

Corollary 7.1. *Let* $\alpha \in [0,1]$. *Then the mapping* r_α *defined on the set of curves as*

$$r_\alpha(f) \ := \ f(0) + \alpha(f(1) - f(0))$$

is a reference point for the Fréchet distance with respect to similarity operations. Its Lipschitz constant is 1.

7.2 Volume of Symmetric Difference

Alt, Blömer, Godau and Wagener [5], and Alt, Fuchs, Rote and Weber [6] investigate the volume of the symmetric difference of compact convex subsets of $\mathbb{R}^d$ as a distance measure in shape matching. For arbitrary compact subsets of $\mathbb{R}^d$ the volume of the symmetric difference is defined as the volume of $A \setminus B$ plus the volume of $B \setminus A$. Alt, Fuchs, Rote and Weber [6] show that the center of mass is a weak reference point, see Definition 1.1, for the symmetric difference for compact convex subsets of $\mathbb{R}^2$ with respect to affine transformations. Its approximation factor is 11/3 and this bound is tight. Weber [51] generalizes this result to higher dimensions and shows that the center of mass is a weak reference point in any dimension $d \geq 2$. Its approximation factor is $1 + 2\frac{d^2}{d+1}$ and again this constant is sharp.

7.3 Volume of Overlap

Substantially different from all distance measures mentioned in this thesis is the volume of overlap. The overlap of two compact subsets A, B in $\mathbb{R}^d$ is defined as the volume of $A \cap B$. It is different because two shapes are the more similar the larger their volume of overlap is. Thus computing the optimum under transformations leads to a maximization problem.

Nevertheless, de Berg, Devillers, van Kreveld, Schwarzkopf and Teillaud [24] remark that minimizing the volume of the symmetric difference of two convex polygons is equivalent to maximizing their volume of overlap. They further show that the translation for which the centers of mass of two convex polygons coincide realizes an overlap that is at least 9/25 of the maximum possible overlap which can be achieved using any translation. They also give an upper bound example where the factor is 4/9 and believe that this is the true bound. De Berg, Devillers, van Kreveld, Schwarzkopf and Teillaud [24] further generalize the approach to arbitrary dimension $d > 2$ and show that matching the centers of mass of two d-dimensional convex polyhedra realizes an overlap volume of at least $(3/(d+3))^d$ times the maximal overlap volume.

7.4 Frobenius Norm

Zikan and Silberberg [54] investigate the minimum Frobenius norm under different classes of transformations. Given two point sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\} \in \mathcal{P}^{d,n}$, the Frobenius norm between these point sets is given by

$$D_F(A,B) \ := \ \min_{\pi \in S_n} \left(\sum_{i=1}^{n} \sum_{j=1}^{d} \left(a_{ij} - b_{\pi(i)j} \right)^2 \right)^{1/2} \ = \ \min_{\pi \in S_n} \left(\sum_{i=1}^{n} \left\| a_i - b_{\pi(i)} \right\|^2 \right)^{1/2}.$$

The Frobenius norm is another distance measure for point sets with the same number of points. Like the bottleneck distance, see Chapter 6, it is computed via the consideration of all possible pairs of points. But unlike the bottleneck distance, it is computed via averaging the squared distances and not just taking the maximum into account. This, in contrast, is similar to the EMD, see Chapter 3.

Zikan and Silberberg [54] use an approach similar to our reference point framework to approximate the minimum Frobenius norm under rigid motions between two point sets of equal size. They first exclude the problem to find an approximate similarity by scaling both point sets in a way that their normalized first moments coincide. They further motivate this approach by showing that this is optimal in the case where the two point sets perfectly match under similarities.

To compute the approximate rigid motion, they first use Algorithm 1.1 based on the centers of mass of the two point sets. We can show that the center of mass is Lipschitz continuous with respect to the Frobenius norm and the proof of Theorem 1.5 carries over, proving a 2-approximation for the Frobenius norm under translations. However, Zikan and Silberberg [54] show that in the case of the Frobenius norm, the translation given by the difference of the two centers of mass is optimal and thus Algorithm 1.1 provides the optimal solution. Note that this is not the case for any other distance measure of shapes considered in this thesis.

Zikan and Silberberg [54] then compute an approximate rotation around the coinciding reference points by using a parametric linear programming approach. To find the optimal rotation for a fixed assignment, they use rotations in the complex plane in the 2-dimensional case, quaternions in 3 dimensions and singular value decomposition for arbitrary dimension d.

It would be interesting to check if an approach similar to the EMD, where one computes the Frobenius norm for all rotations aligning at least one point of each set, leads to a constant-factor approximation.

Bibliography

[1] Pankaj K. Agarwal and Jeff M. Phillips. On bipartite matching under the rms distance. In *Proc. 18th Canadian Conf. on Comput. Geom.*, pages 143–146, 2006.

[2] Pankaj K. Agarwal, Micha Sharir, and Sivan Toledo. Applications of parametric searching in geometric optimization. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 72–82, 1992.

[3] H. Alt, O. Aichholzer, and Günter Rote. Matching shapes with a reference point. *Internat. J. Comput. Geom. Appl.*, 7:349–363, 1997.

[4] H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 186–193, 1991.

[5] H. Alt, J. Blömer, M. Godau, and H. Wagener. Approximation of convex polygons. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, volume 443 of *Lecture Notes Comput. Sci.*, pages 703–716. Springer-Verlag, 1990.

[6] H. Alt, U. Fuchs, Günter Rote, and G. Weber. Matching convex shapes with respect to the symmetric difference. In *Algorithms — Proc. 4th Annu. European Sympos. Algorithms*, volume 1136 of *Lecture Notes Comput. Sci.*, pages 320–333. Springer-Verlag, 1996.

[7] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5:75–91, 1995.

[8] Helmut Alt, Christian Knauer, and Carola Wenk. Matching polygonal curves with respect to the fréchet distance. In *Proc. 18th Int. Sympos. on Theoretical Aspects of Comp. Science*, pages 63–74, 2001.

[9] N. Amenta. Bounded boxes, Hausdorff distance, and a new proof of an interesting Helly theorem. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 340–347, 1994.

[10] I. Assent, A. Wenning, and T. Seidl. Approximation techniques for indexing the earth mover's distance in multimedia databases. In *Proc. 22nd Int. Conf. on Data Engineering (ICDE)*, 2006.

[11] M. J. Atallah. A linear time algorithm for the Hausdorff distance between convex polygons. *Inform. Process. Lett.*, 17:207–209, 1983.

[12] Wolfgang W. Bein, Peter Brucker, James K. Park, and Pramod K. Pathak. A monge property for the d-dimensional transportation problem. *Discrete Appl. Math.*, 58(2):97–109, 1995.

[13] Marc Benkert, Alexander Wolff, Florian Widmann, and Takeshi Shirabe. The minimum Manhattan network problem: Approximations and exact solution. *CGTA*, 2006.

[14] Manuel Blum, Robert W. Floyd, Vaughan R. Pratt, Ronald L. Rivest, and Robert Endre Tarjan. Time bounds for selection. *J. Comput. Syst. Sci.*, 7(4):448–461, 1973.

[15] S. Cabello, P. Giannopoulos, C. Knauer, and G. Rote. Matching point sets with respect to the earth mover's distance. In *Proc. European Sympos. Algorithms*, 2005.

[16] Victor Chepoi, Karim Nouioua, and Yann Vaxes. A rounding algorithm for approximating minimum manhattan networks. 2005.

[17] L. P. Chew, M. T. 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.

[18] Kenneth L. Clarkson. Nearest-neighbor searching and metric space dimensions. In Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk, editors, *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*, pages 15–59. MIT Press, 2006.

[19] Scott Cohen. *Finding Color and Shape Patterns in Images.* PhD thesis, Stanford University, Department of Computer Science, 1999.

[20] Scott Cohen and Leonidas Guibas. The earth mover's distance: Lower bounds and invariance under translation. Technical Report CS-TR-97-1597, 1997.

[21] Scott. D. Cohen and Leonidas. J. Guibas. The earth mover's distance under transformation sets. In *Proc. 7th IEEE Int. Conf. Comp. Vision*, pages 173–187, 1999.

[22] R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *J. ACM*, 34(1):200–208, 1987.

[23] I. K. Daugavet. Some applications of the marcinkiewicz-berman identity. *Vestnik Leningrad Univ., Math.*, pages 321–327, 1974.

[24] Mark de Berg, Olivier Devillers, Marc van Kreveld, Otfried Schwarzkopf, and Monique Teillaud. Computing the maximum overlap of two convex polygons under translations. In *Proc. 7th Annu. Internat. Sympos. Algorithms Comput.*, volume 1178 of *Lecture Notes Comput. Sci.*, pages 126–135. Springer-Verlag, 1996.

[25] A. Efrat, A. Itai, and M. J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31:1–28, 2001.

[26] Panos Giannopoulos. *Geometric Matching of Weighted Point Sets.* PhD thesis, Universiteit Utrecht, 2005.

[27] Panos Giannopoulos and Remco C. Veltkamp. A pseudo-metric for weighted point sets. In *Proc. 7th Europ. Conf. on Comp. Vision, LNCS 2352*, pages 715–731, 2002.

[28] M. Godau. *On the Complexity of Measuring the Similarity Between Geometric Objects in Higher Dimensions.* PhD thesis, Freie Universität Berlin, 1998.

[29] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[30] K. Graumann and T. Darell. Fast contour matching using approximate earth mover's distance. In *Proc. 1991 IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition*, pages 220–227, IEEE Service Center, Piscataway, NJ, USA (IEEE cat n 91CH2983-5), 2004. IEEE.

[31] B. Grünbaum. *Convex Polytopes*. John Wiley & Sons, New York, NY, 1967.

[32] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Approximating a minimum Manhattan network. *Nordic J. Comput.*, 8:219–232, 2001.

[33] Joachim Gudmundsson, Oliver Klein, Christian Knauer, and Michiel Smid. Small manhattan networks and algorithmic applications for the earth mover's distance. In *Proc. 23rd Europ. Workshop Comp. Geom.*, pages 174–177, 2007.

[34] J. D. Horton. Sets with no empty convex 7-gons. *Canad. Math. Bull.*, 26:482–484, 1983.

[35] D. P. Huttenlocher and K. Kedem. Computing the minimum Hausdorff distance for point sets under translation. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 340–349, 1990.

[36] Piotr Indyk. A near linear time constant factor approximation for euclidean bichromatic matching (cost), to appear. In *Proc. 18th Symp. on Disc. Alg.*, 2007.

[37] Oliver Klein and Remco C. Veltkamp. Approximation algorithms for the earth mover's distance under transformations using reference points. In *Proc. 16th Intl. Sympos. Algorithms and Computation*, pages 1019–1028, 2005.

[38] Oliver Klein and Remco C. Veltkamp. Approximation algorithms for the earth mover's distance under transformations using reference points. Report B 05-11, Freie Universität Berlin, Institut für Informatik, 2005.

[39] Christian Knauer. *Algorithms for Comparing Geometric Patterns*. PhD thesis, Freie Universität Berlin, Institut für Informatik, 2002.

[40] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983.

[41] J. B. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations Research*, 41:338–350, 1993.

[42] K. Przesławski and D. Yost. Continuity properties of selectors and michael's theorem. *Michigan Math. J.*, 36:113–134, 1989.

[43] S. Rachev and L. Rüschendorf. *Mass Trasportation Problems. Vol. I: Theory, Vol. II: Applications*. Springer, 1998.

[44] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. of Comp. Vision*, 40:99–121, 2000.

[45] D. Rutovitz. Some parameters associated with a finite dimensional banach space. *J. London Math. Soc.*, 2:241–255, 1965.

[46] R. Schneider. Krümmungsschwerpunkte konvexer Körper. *Abh. Math. Sem. Hamburg*, 37:112–132, 1972.

[47] G. C. Shepard. The Steiner point of a convex polytope. *Canad. J. Math.*, 18:1294–1300, 1966.

[48] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. Oostrum. Using transportation distances for measuring melodic similarity. In *Proc. 4th Int. Conf. on Music Information Retrieval*, pages 107–114, 2003.

[49] P. M. Vaidya. Geometry helps in matching. *SIAM J. Comput.*, 18:1201–1225, 1989.

[50] C. Villani. *Topics in Optimal Transportation.* American Mathematical Society, 2003.

[51] Gerald Weber. The centroid is a reference point for the symmetric difference in d dimensions. Report UoA-SE-2004-1, University of Auckland, 2004.

[52] Carola Wenk. *Shape Matching in Higher Dimensions.* PhD thesis, Freie Universität Berlin, Institut für Informatik, 2002.

[53] A. C. Yao. Space-time trade-off for answering range queries. In *Proc. 14th Annu. ACM Sympos. Theory Comput.*, pages 128–136, 1982.

[54] Karel Zikan and Theresa M. Silberberg. The Frobenius metric in image registration. In L. Shapiro and A. Rosenfeld, editors, *Computer Vision and Image Processing*, pages 385–420. Academic Press, 1992.

Appendix A

Lower Bound for the EMD Under Rotations

l	Max 1: r_1^*	Max 2: r_2^*	$\sin\phi$	Approx. Ratio
0.1	0.2744646880	3.643455948	0.7961286352	1.031102773
0.2	0.3808573470	2.625655007	0.7454811025	1.054514795
0.3	0.4648467435	2.151246651	0.6914893705	1.074950280
0.4	0.5388130560	1.855931271	0.6310736355	1.093317292
0.5	0.6077079156	1.645527357	0.5619979981	1.109842515
0.6	0.6747641047	1.481999402	0.4818463712	1.124449510
0.7	0.7432426857	1.345455555	0.3875854431	1.136840155
0.8	0.8174753427	1.223278487	0.2755283240	1.146487880
0.82	0.8334935771	1.199769293	0.2506916271	1.148025105
0.84	0.8500073125	1.176460467	0.2250295597	1.149416170
0.86	0.8670512485	1.153334364	0.1985673307	1.150654512
0.88	0.8846470340	1.130394340	0.1713574339	1.151734043
0.9	0.9027971779	1.107668505	0.1434865666	1.152649523
0.92	0.9214788335	1.085212122	0.1150812464	1.153397032
0.94	0.9406388679	1.063107250	0.08631006634	1.153974446
0.96	0.9601920648	1.041458305	0.05738036238	1.154381897
0.98	0.9800241231	1.020383046	0.02852788354	1.154622116
0.99	0.9900030114	1.010097938	0.01420800295	1.154681108
0.992	0.9920015410	1.008062950	0.01135649035	1.154688126
0.994	0.9940006497	1.006035560	0.008509696885	1.154693569
0.996	0.9960001923	1.004015870	0.005667860612	1.154697446
0.998	0.9980000240	1.002003984	0.002831217249	1.154699765
0.999	0.9990000030	1.001000998	0.001414915889	1.154700343
0.9992	0.9992000015	1.000800639	0.001131820951	1.154700415
0.9994	0.9994000006	1.000600360	0.0008487816639	1.154700466
0.9996	0.9996000002	1.000400160	0.0005657982564	1.154700516
0.9998	0.9998000000	1.000200040	0.0002828709586	1.154700534
0.9999	0.9999000000	1.000100010	0.0001414284225	1.154700537

l	Max 1: r_1^*	Max 2: r_2^*	$\sin\phi$	Approx. Ratio
1.0002	0.9998000400	1.000200000	0.0002828143900	1.154700534
1.0004	0.9996001601	1.000400000	0.0005655719826	1.154700523
1.0006	0.9994003604	1.000599999	0.0008482725490	1.154700472
1.0008	0.9992006410	1.000799998	0.001130915861	1.154700418
1.0010	0.9990010020	1.000999997	0.001413501690	1.154700349
1.002	0.9980040159	1.001999976	0.002825560636	1.154699786
1.004	0.9960161263	1.003999808	0.005645237041	1.154697457
1.006	0.9940364234	1.005999354	0.008458804659	1.154693654
1.008	0.9920649966	1.007998471	0.01126604216	1.154688338
1.010	0.9901019326	1.009997019	0.01406673160	1.154681499
1.02	0.9804148885	1.019976351	0.02796458599	1.154625183
1.04	0.9617093292	1.039815222	0.05515506822	1.154406259
1.06	0.9439343170	1.059395746	0.08140244953	1.154055573
1.08	0.9271092368	1.078621548	0.1065859353	1.153585891
1.10	0.9112253996	1.097423316	0.1306321495	1.153010297
1.12	0.8962521756	1.115757403	0.1535092660	1.152341679
1.14	0.8821438222	1.133601999	0.1752186327	1.151592317
1.16	0.8688456918	1.150952361	0.1957860061	1.150773650
1.18	0.8562991849	1.167816130	0.2152537015	1.149896121
1.2	0.8444453224	1.184209295	0.2336742275	1.148969132
1.3	0.7937000871	1.259921747	0.3121207701	1.143853002
1.4	0.7533748881	1.327360409	0.3727553473	1.138401676
1.5	0.7200219492	1.388846550	0.4207934694	1.132983698
1.6	0.6916060430	1.445909865	0.4597398903	1.127771402
1.7	0.6668707176	1.499541026	0.4919424548	1.122838338
1.8	0.6449941469	1.550401666	0.5190129196	1.118208435
1.9	0.6254094904	1.598952391	0.5420895866	1.113880805
2.0	0.6077079156	1.645527357	0.5619979981	1.109842514

Zusammenfassung

Die geometrische Mustererkennung ist ein wichtiges Thema unter anderem in der algorithmischen Geometrie, beim computerunterstützten Sehen und in der Robotik, um nur einige Anwendungsgebiete zu nennen. Für eine feste Abstandsfunktion und eine feste Klasse von Transformationen $\mathcal{T}$ kann das Problem wie folgt beschrieben werden: Gegeben seien zwei Muster A und B, finde eine Transformation $T^* \in \mathcal{T}$, so dass der Abstand zischen A und $T^*(B)$ minimal ist unter allen Transformationen $T \in \mathcal{T}$. Die betrachteten Muster in dieser Arbeit sind im Wesentlichen kompakte Teilmengen, gewichtete Punktmengen, Punktmengen mit einer festen Anzahl von Punkten und Wahrscheinlichkeitsmaße. Unabhängig von der Wahl einer dieser speziellen Muster ist das Finden einer optimalen Transformation aufwendig, falls überhaupt möglich. Daher konzentrieren wir uns auf Näherungsalgorithmen.

Der Ansatz dieser Arbeit basiert auf Referenzpunkten. Dieser Ansatz wurde unter anderem von Alt, Behrends und Blömer [4] und Alt, Aichholzer und Rote [3] gewählt. Die Autoren dieser beiden Arbeiten benutzen spezielle Abbildungen, sogenannte Referenzpunkte, um eine relative Position der beiden Muster zueinander zu bestimmen. Dieses reduziert die Freiheitsgrade des zugrundeliegenden Problems um die Dimension des Raumes. Ein Referenzpunkt ist eine Lipschitz-stetige Funktion definiert auf der Menge der Muster, die in den $\mathbb{R}^d$ abbildet. Zusätzlich bleibt die relative Position des Referenzpunktes bei Anwendung einer Transformation erhalten.

In dieser Dissertation beschäftigen wir uns mit Näherungsalgorithmen für die Mustererkennung bezüglich verschiedener Abstandsmaße. Im Wesentlichen sind dies der Hausdorff-Abstand, der diskrete und kontinuierliche Transportabstand und der Engpass-Abstand. Dabei werden Näherungsalgorithmen für die verschiedenen Maße bezüglich Verschiebungen, starren Bewegungen, d.h. Verschiebungen und Drehungen, und positiven Ähnlichkeitsabbildungen, d.h. starren Bewegungen in Verbindung mit positiven Streckungen, betrachtet.

Die Grundstruktur der Näherungsalgorithmen für die verschiedenen Abstandsmaße und unterschiedlichen Transformationen ist jeweils ähnlich und wird in einem allgemeinen Rahmen beschrieben. Zuerst wird die relative Position der beiden Muster zueinander durch Berechnung der Referenzpunkte der beiden Mengen bestimmt. Anschließend werden die Muster so verschoben, dass die beiden Referenzpunkte übereinstimmen. Im Anschluss bestimmen wir eine Drehung für eine der beiden Mengen, sodass der Abstand der beiden Mengen, den Wert einer optimalen Lösung höchstens um einen konstanten Faktor übersteigt. Streckungen werden betrachtet, indem vor dem Finden einer Drehung eine Streckung bestimmt wird, die einer optimalen Streckung sehr nahe kommt.