# ABSTRACT

Title : APPROXIMATION ALGORITHMS FOR
POINT PATTERN MATCHING AND SEARCHING

Minkyoung Cho, Doctor of Philosophy, 2010

Directed by: Professor David M. Mount
Department of Computer Science

Point pattern matching is a fundamental problem in computational geometry. For given a reference set and pattern set, the problem is to find a geometric transformation applied to the pattern set that minimizes some given distance measure with respect to the reference set. This problem has been heavily researched under various distance measures and error models.

Point set similarity searching is variation of this problem in which a large database of point sets is given, and the task is to preprocess this database into a data structure so that, given a query point set, it is possible to rapidly find the nearest point set among elements of the database. Here, the term *nearest* is understood in above sense of pattern matching, where the elements of the database may be transformed to match the given query set. The approach presented here is to compute a low distortion embedding of the pattern matching problem into an (ideally) low dimensional metric space and then apply any standard algorithm for nearest neighbor searching over this metric space.

This main focus of this dissertation is on two problems in the area of point pat-

tern matching and searching algorithms: (1) improving the accuracy of alignment-based point pattern matching and (2) computing low-distortion embeddings of point sets into vector spaces.

For the first problem, new methods are presented for matching point sets based on alignments of small subsets of points. It is shown that these methods lead to better approximation bounds for alignment-based planar point pattern matching algorithms under the Hausdorff distance. Furthermore, it is shown that these approximation bounds are nearly the best achievable by alignment-based methods.

For the second problem, results are presented for two different distance measures. First, point pattern similarity search under translation for point sets in multidimensional integer space is considered, where the distance function is the symmetric difference. A randomized embedding into real space under the $L_1$ metric is given. The algorithm achieves an expected distortion of $O(\log^2 n)$. Second, an algorithm is given for embedding $\mathbb{R}^d$ under the Earth Mover's Distance (EMD) into multidimensional integer space under the symmetric difference distance. This embedding achieves a distortion of $O(\log \Delta)$, where $\Delta$ is the diameter of the point set. Combining this with the above result implies that point pattern similarity search with translation under the EMD can be embedded into real space in the $L_1$ metric with an expected distortion of $O(\log^2 n \log \Delta)$.

# APPROXIMATION ALGORITHMS

# FOR

# POINT PATTERN MATCHING AND SEARCHING

by

Minkyoung Cho

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:
Professor David M. Mount, Chair/Advisor
Professor Samir Khuller
Professor David W. Jacobs
Professor Amitabh Varshney
Professor Mark A. Austin, Dean's Representative

# Dedication

To my parents and hime

# Acknowledgments

I heartily thank my advisor, Professor David Mount, whose continual encouragement, guidance, and support enabled me to complete my Ph.D. He is among the best teachers that I have ever met. He provided a role model for the personality a teacher should have and how to guide students. He attended to me like a father. Whenever I experienced personal or academic difficulties, he advised me with consideration and sympathy. I remember the first time that we first met. He listened to my words and happily discussed some possible research topics, even though I was novice in research. He was professional in his approach to research. Whenever solving a problem, he would encourage me to obtain better results. He assisted me in the writing of my dissertation. He would sometimes pick up a red pen and proceed to revise my writing. After finishing it, the paper became red all around. I will not forget the time that I spent with him. My time in the Ph.D program was happy and joyful. He is amazingly nice, I would want to spend more time with him. I am so sorry that I must leave to take the next step in my career. I am very thankful for the time we have spent together.

I am deeply thankful to my committee members: Professor Samir Khuller, Professor David M. Jacobs, Professor Amitabh Varsheny, and Professor Mark Austin. Occasionally Professor Samir Khuller dropped by my office and, with kindness and generosity, he would ask me how I was doing. Whenever I missed some procedures, his brief and concise emails enlightened me. Also, his excellent teaching in the three courses I took from him solidified my knowledge of algorithms. My research

discussions with Professor David W. Jacobs were both very helpful and insightful. He is a distinguished researcher with a well-established background. His class was full of enthusiasm and greatly helped to improve my research. Professor Amitabh Varshney and Professor Mark Austin happily served on my thesis committee, and I thank them for their invaluable time in reviewing my dissertation.

I was so fortunate to have many great and wonderful people around me during my time in the graduate program.

I specially thank my lab colleagues, Nargess Memarsadeghi, Guilherme Fonseca, Sorelle Friedler, and Eunhui Park. They attended my practice talks and provided many valuable comments. I had weekly research meetings and discussions with the members of Korean Graduate Students Vision and AI Research Group (KGVISA) and the Korean Graduate Students System Research Group (KGSYS). Through these meetings, I expanded my knowledge of various areas. I want specially to acknowledge their members: (KGVISA) Kyongil Yoon, Hyoungjune Yi, Bohyung Han, Kyungnam Kim (KGSYS) Hyeonsang Eom, Joon-Hyuk yoo, Jihwang Yeo, Minho Shin, Soobum Lee, Seungjoon Lee, Sunghyun Chun, Yoo Ah Kim, Jae Hwan Lee, Ji Sun Shin, and Jinhyuk Jung.

I am very grateful to my close friends, mostly in the Computer Science Department: Il-Chul Yoon, Heejong Sung, Jik-soo Kim, Boram Lee, Youngmin Kim, Jiyoung Kim, Beomseok Nam, Suryoun Jung, Doon Hoon Park, Sangchul Song, Inseok Choi, Sukhyun Song, Sungwoo Park, Hyungyoung Song, Adam Bender, Wontaek Seo, Joonghoon Lee, Hyejung Lee, Hyungtae Cho, Youngho Cho, Hyungtae Lee, Tak Yeon Lee, Alex Aris, Dov Gordon, and Ryan Farrell. Because of them, my

school life was very rich with happiness and well-being. In particular, I would like to express my special gratitude to Il-Chul Yoon and Sangchul Song, who enjoyed and shared their thoughts and knowledge about job searching and my dissertation as well as life in general. Finally, I thank God for the presence of Angela Song-Ie Noh, Jee Hye Han, and Hyuk Oh. Their emotional support contributed to my great joy of life.

I finish this acknowledgement by mentioning my family. I am forever indebted to my parents and family. I feel deep gratitude to my parents, sister and brother for their endless love throughout my life.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Geometric point pattern matching problem is a fundamental computational problem and has numerous applications in areas such as computer vision [47], image and video compression [5], model-based object recognition [33], and computational chemistry [25].

The most common formulation considered in computational geometry involves determining the degree of similarity between two given point sets, subject to some group of allowable geometric transformations. For example, for planar point sets the transformation group might consist of translations, rigid motions (translation and rotation), similarities (translation, rotation, and uniform scaling), and affine transformations (translation, rotation, nonuniform scaling, and shearing). We assume that similarity (or more accurately, dissimilarity) is measured by some *distance function* that maps two point sets to a nonnegative real. There are many different ways in which distance between two point sets can be measured, as we shall see later. We can characterize the notion of distance between two point sets under an arbitrary group of transformations as follows.

**Definition 1** Let $P$ and $Q$ be two finite point sets in $\mathbb{R}^d$, let $\mathcal{T}$ be a group of

geometric transformations, $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d$, and let *dist* be some distance function between point sets. Define *Dist* to be corresponding distance function under the transformations of $\mathcal{T}$, that is:

$$Dist(P, Q) = \min_{t \in \mathcal{T}} dist(tP, Q).$$

$\diamondsuit$

Depending on what distance measure is used and what type of errors are to be tolerated, problem solving techniques are quite different. There are a number of key criteria that characterize the exact nature of a point pattern matching problem.

**Exact versus Noisy Data:** Point data may arise from many different sources. Exact data arise from a discrete source, which may be assumed to be free of measurement error. Early research on the algorithmic complexity of point pattern matching focused on these instances, largely because they are the easiest to deal with. In most applications, however, point data arises from measurements of continuous data and hence are subject to errors of various sources, such as sensing errors and discretization.

**Partial or full matching:** In some applications, it is desirable to locate instances of a small pattern set in a larger reference set. In other applications (arising for example from image registration [10]) it is assumed that the two point sets are of roughly equal size and the objective is to find the transformation that most nearly aligns the two sets.

**1-to-1 versus Many-to-1 Correspondences and Robustness:** It is often de-

sirable to consider 1-to-1 matches between point sets. This is particularly true if it is known that there are no missing or spurious point in the sets, and hence each point in the reference set may be assumed to correspond to a single point in the pattern set. In many applications, however, there may be missing and spurious points. These arise as a consequence of obscuration or errors in feature-point selection.

**Objective Function:** There are many ways of formally defining the degree of similarity between two point sets. When data is exact, it is common to consider measures such a symmetric difference, which count the number of points of one set that are not in the other. When noisy data is presented, other measures based on the distances between nearby points, such as the Hausdorff distance [2,16,24,53,54], bottleneck distance [23], and earth mover's distance [18] are often used. (Definitions will be presented below.)

**Single pair versus database search:** In applications like point-based image registration [35, 36, 58], a single pair of point sets is given, and the goal is to find an optimal aligning transformation. In the database search problem, it is assumed that a large collection of sets is given, and the problem involves processing this collection into a data structure so that, given a query point set, it is possible to identify similar point sets.

In Chapter 2 a more extensive review of the literature in this area is presented.

## 1.1 Summary of Results

The results of this dissertation are focused on three different topics. The first involves improvements to an existing approximation algorithm for point pattern matching under the Hausdorff distance (defined below). The second involves the design of a new approach for point pattern similarity search based on computing a low-distortion embedding of the point pattern matching problem into a vector space. The third involves of an embedding algorithm for the earth mover's distance and point pattern similarity searching under the earth mover's distance.

### 1.1.1 Improved Approximation Bounds for Planar Point Pattern Matching

Given point sets $P$ and $Q$ the *directional Hausdorff distance* (also called the *directional Hausdorff distance*), denoted $h(P, Q)$, is defined to be

$$h(P, Q) = \max_{p \in P} \min_{q \in Q} \|pq\|,$$

where $\|pq\|$ denotes the Euclidean distance between points $p$ and $q$.

The directional Hausdorff distance is natural in applications where the pattern set $P$ is expected to match some subset of the background set $Q$. In applications where it is desired that every point of $P$ matches some point of $Q$ and vice versa, a bidirectional similarity measure may be more appropriate such as the *bidirectional Hausdorff distance*, which is defined to be $\max(h(P, Q), h(Q, P))$. When matching with bidirectional similarity measures, it may be possible to identify a global

reference point (such as the centroid of each set) about which to anchor the alignment. This is not possible, however, under the directional Hausdorff distance, since such statistics may be distorted by unmatched outlying points. Throughout, unless otherwise specified, we use the term *Hausdorff distance* to denote the directional Hausdorff distance.

A simple and natural approach is to consider transformations induced by aligning a small subset of points from one set to the other. This is arguably the simplest and most easily implemented algorithm for approximate pattern matching, and it is the basis of some of the most popular methods in computer vision, such as RANSAC [26]. Goodrich, Mitchell, and Orletsky [28] were the first to prove an upper bound on the approximation ratio such a simple alignment-based algorithms. They considered point pattern matching under a number of different transformation spaces and in different dimensions. For the case of rigid transformations in the plane, their algorithm computes a diametrical pair for $P$ and then computes for every pair of distinct points of $Q$ a rigid transformation that aligns these pairs. It then returns the transformation achieving the minimum Hausdorff distance. Their algorithm runs in $O(n^2 m \log n)$ time. They prove that it returns an aligning transformation whose Hausdorff distance is at most a factor of 4 larger than the optimum Hausdorff distance.

We considered this problem, with the objective of improving the approximation ratio of the Goodrich, Mitchell, Orletsky algorithm, while retaining the same simple algorithmic structure. Their algorithm is based on aligning points one by one, and henceforth we refer to this as *serial alignment*. We show that it is possible to improve

on their approximation ratio of 4. Our approach has the same running time as theirs and, like theirs, is very easy to implement. It is based on a minor modification that selects the transformation that best aligns the entire subset of points, which we call *symmetric alignment*. Let $A_{ser}$ and $A_{sym}$ denote the approximation ratios for these respective algorithms.

Rather than just considering the worst-case approximation ratios, we analyzed the approximation ratios of these algorithms in a manner that is sensitive to the optimal Hausdorff distance. For each problem instance $P$ and $Q$, we define an geometric parameter $\rho$, called the *distance ratio*, to be half the ratio of the diameter of $P$ to the optimum Hausdorff distance between $P$ and $Q$. We showed that, as the distance ratio increases, the accuracy of the approximation increases as well. We feel that this analysis is useful because large values of $\rho$ often arise in applications. For example, in document analysis and satellite image analysis, the ratio of the diameter of a typical pattern ranges from tens to hundreds of pixels, while the expected digitization error is on the order of a single pixel. Let $A_{ser}(\rho)$ and $A_{sym}(\rho)$ denote the approximation ratios for serial and symmetric alignment, respectively, as a function $\rho$.

Our results from [16] show that, for serial alignment, the approximation bound $A_{ser}$ satisfies

$$c_\infty + \frac{1}{27\rho^2} \;\leq\; A_{ser}(\rho) \;\leq\; c_\infty + \frac{9}{4\rho},$$

where $c_\infty \approx 3.19$. For symmetric alignment, the approximation bound $A_{sym}$ satisfies

$$3 + \frac{1}{10\rho^2} \;\leq\; A_{sym}(\rho) \;\leq\; 3 + \frac{1}{\sqrt{3}\,\rho}.$$

Observe that the approximation ratio for symmetric alignment is better than that of serial alignment for almost all but very small values of $\rho$. Further, for (typical) applications where distance ratio is large, the approximation factor of symmetric alignment is close to 3.

Our results can also be applied to provide a modest improvement in the running time of the $\varepsilon$-approximation algorithm of Indyk *et al.* [39]. Their algorithm uses the simple alignment algorithm as a subroutine. The running time of their algorithm has a cubic dependence on the (upper bound on the) approximation ratio of the alignment algorithm. So, improving the approximation ratio bound by a factor of $f$ results in factor of $f^3$ reduction in the running time of their algorithm.

These results are published in [16]. Complete details of the algorithm results can be found in Chapter 3.

### 1.1.2 Similarity Search for Point Sets with Translation under Symmetric Difference

When dealing with a large database of point sets, an important problem is how to perform similarity search, that is, to find the closest point set of the database to a given query pattern. The objective is to preprocess the elements of the database so that searches can be answered efficiently.

We considered this problem in a relatively simple context, but one that still leads to quite an interesting computational problem. We assume that point sets have integer coordinates, that they are to be matched subject to an unknown translation,

and that there is a significant fraction of outliers, that is, points from one set may not match any point of the other set. We assume, however, corresponding points (subject to the optimum translation) match identically. (This is to be contrasted with measures such as the partial Hausdorff distance [35], where both outliers and near misses are tolerated.) Outliers are challenging because global properties of the point sets, based for example on the identification of reference points such as centroids [2], are not applicable. The distance metric we use is the size of the symmetric difference of the two point sets, which is to be minimized through some translation of one set relative to the other. (Formal definitions are given below.)

The approach presented here is based on finding a function that maps a point set from one metric space to another. A *metric space* $(X, d)$ is a set $X$, and a nonnegative distance function $d$ such that for all $x, y, z \in X$, (1) $d(x, x) = 0$, (2) $d(x, y) = d(y, x)$, and (3) $d(x, z) \leq d(x, y) + d(y, z)$. The last condition is the *triangle inequality*. The distortion of such an embedding function is defined to be the maximum multiplicative variation that distances might suffer in the mapping process. More precisely, given two metric spaces $(X, d_X)$ and $(Y, d_Y)$, we say that an embedding $f : X \to Y$ has *distortion* $c$, if there exist $c_1 c_2 = c$ such that for all $x, x' \in X$,

$$\frac{1}{c_1} d_X(x, x') \;\leq\; d_Y(f(x), f(x')) \;\leq\; c_2 d_X(x, x').$$

Our approach is to embed the points of the database into a metric space for which there exists an efficient similarity search algorithm. Given any query point set, similarity search is performed by a reduction to nearest neighbor searching in

the metric space.

Consider a point set consisting of at most $n$ points on the $d$-dimensional integer grid, where $d$ is a constant. We assume that the coordinates of each point are bounded above by a polynomial function of $n$. As usual, let $\mathbb{Z}$ denote the set of integers, and let $\mathbb{Z}_u$ denote $\{0, 1, 2, \ldots, u-1\}$. (We do not assume that $u$ is prime.) Let $\mathbb{Z}^d$ denote the set of $d$-element vectors over $\mathbb{Z}$ and define $\mathbb{Z}_u^d$ analogously for $\mathbb{Z}_u$. Let $\mathbb{Z}_u^d(\leq n)$ denote the collection of point sets over $\mathbb{Z}_u^d$ that contain at most $n$ points. Given two finite sets $P$ and $Q$, let $P \ominus Q$ denote their *symmetric difference*, that is,

$$P \ominus Q \;=\; (P \setminus Q) \cup (Q \setminus P).$$

The cardinality of the symmetric difference is a well known metric on finite sets, which we denote by $|P \ominus Q|$.

Given a point set $P$ and any $t \in \mathbb{Z}^d$, the *translate* $P + t$ is defined to be $\{p + t \mid p \in P\}$. Extending the symmetric difference, we define the *symmetric difference distance under translation*, denoted $\langle P \ominus Q \rangle$, to be

$$\langle P \ominus Q \rangle \;=\; \min_{t \in \mathbb{Z}^d} |(P + t) \ominus Q| \,.$$

It is easy to verify that this is a metric. (See Lemma 4.2.1.) Throughout, we will assume that $P$ and $Q$ are taken from $\mathbb{Z}_u^d(\leq n)$.

Let $\ell_1^d$ denote the metric space consisting of real $d$-dimensional space $\mathbb{R}^d$ endowed with the $L_1$ metric. Given $x, y \in \ell_1^d$, we denote their $L_1$ distance by $\|x - y\|_1$. We use the terms *randomized embedding* and *randomized function* throughout to denote a function computed by a randomized algorithm that satisfies the given

probability bounds. We also use log to denote logarithm base 2 and ln to denote the natural logarithm. In Chapter 4 we show that such a translation-invariant embedding is possible, by proving the following theorem.

**Theorem 1** *Given integers $n$ and $u$, where $u \leq n^{O(1)}$, a constant $d$, and failure probability $\beta$, there exists a randomized embedding $\Psi : \mathbb{Z}_u^d(\leq n) \rightarrow \ell_1^m$, where $m = O(n \log^2 n \log(1/\beta))$ such that for any $P, Q \in \mathbb{Z}_u^d(\leq n)$:*

*(i)* $\|\Psi P - \Psi Q\|_1 \leq (2 \log n) \langle P \ominus Q \rangle$.

*(ii)* $\|\Psi P - \Psi Q\|_1 \geq \frac{1}{17 \log n} \langle P \ominus Q \rangle$, *with probability at least* $1 - \beta$, *and*

*This embedding can be computed in time $O(n \log^4 n \log(1/\beta))$.*

Note that part (i) of the above theorem holds irrespective of randomization. It follows that the resulting embedding achieves a distortion of at most $34 \log^2 n$, with probability at least $1 - \beta$.

This result shows how to embed point sets under translation into $L_1$ space. Since each point set is mapped into a point in this space, similarity search under translation can be reduced to (approximate) nearest neighbor searching among the embedded points. Any standard method for (approximate) nearest neighbor searching may be applied [6, 7, 38, 43].

The principal shortcoming of the above result is that the dimension of the space into which the points are embedded is superlinear in $n$. The following result shows that the dimension can be reduced to a quantity that grows only logarithmically in $n$. The price that we pay is that the distortion bounds hold in expectation only. The expected distortion is $O(\log^2 n)$.

**Theorem 2** *Given positive integers $n$ and $u$, where $u \leq n^{O(1)}$, and a constant $d$, there exists a randomized embedding $\Psi' \colon \mathbb{Z}_u^d(\leq n) \to \ell_1^m$, where $m = O(\log n)$, such that for any two sets $P, Q \in \mathbb{Z}_u^d(\leq n)$:*

$$(i)\ \mathrm{E}\left[\|\Psi' P - \Psi' Q\|_1\right] \ \leq\ (3 \log n)\, \langle P \ominus Q \rangle.$$

$$(ii)\ \mathrm{E}\left[\|\Psi' P - \Psi' Q\|_1\right] \ \geq\ \tfrac{1}{17 \log n}\, \langle P \ominus Q \rangle$$

*This embedding can be computed in expected time $O(n \log^4 n)$.*

Complete details of our algorithm and results can be found in Chapter 4. These results are published in [15].

### 1.1.3 Similarity Search for Point Sets with Translation under EMD

The Earth-Mover's Distance (EMD) [3, 18] is a well-known distance measure, with many applications in computer vision and image processing. The concept was introduced as a means of describing the distance between two probability distributions as a function of the effort needed to convert one into the other. This definition can be applied to a pair of finite point sets $P$ and $Q$ of equal cardinality. Given two point sets $P$ and $Q$ of equal cardinality, let $\mathcal{M}$ denote the set of bijections from $P$ to $Q$. Then, the *earth mover's distance* between $P$ and $Q$ is defined to be

$$\mathrm{EMD}\,(P, Q) \ =\ \min_{M \in \mathcal{M}} \sum_{(p,q) \in M} \|p - q\|_2\,.$$

Computing the EMD between two point sets reduces to computing a minimum weight perfect matching in the complete bipartite graph $(P, Q, P \times Q)$, where the

weight of each edge $(p, q)$ is the Euclidean distance between these points.

We consider the point pattern similarity search problem under the EMD metric. The trivial solution would be to compute the EMD from the query point set to each point set of the collection and return the closest. However, this brute-force search is not acceptable in cases where the total number of point sets is very large. To support efficient searching, a well-formed index or canonical form of a point set is necessary. The approach given here is to embed a point set under EMD into a low-dimensional vector space under some well-known metric space, so that we can use some well-known data structures such as the (approximately) nearest neighbor search algorithms.

The problem of computing low distortion embeddings of the EMD into has been studied by Indyk and Thaper [40], Shirdhonkar and Jacobs [55], and Wang *et al.* [60].

Indyk and Thaper designed an algorithm for embedding EMD in $\mathbb{R}^d$ into $\ell_1$ in $\mathbb{Z}^{d'}$, where $d'$ depends on the cardinality $n$ and the spread (diameter) $\Delta$. Under the assumption that the minimum distance between any two points is at least 1, they showed that the distortion is $O(\log \Delta)$. By merging the embedding results and Locality Sensitive Hashing (LSH), they designed an algorithm for point pattern search under EMD metric [40].

Shirdhonkar and Jacobs presented an linear time algorithm to compute a wavelet-based variant of the EMD. They show that their measure is a metric, and that it is (approximately) equivalent to the EMD in the sense that the ratio of the EMD to the wavelet EMD is bounded within some constant [55]. These algorithms

provide efficient solutions to similarity search for point sets under EMD.

We can naturally extend the problem to apply to the point sets with translations. Given two finite point sets $P$ and $Q$ of equal cardinality, let $\mathcal{M}$ denote the set of bijections from $P$ to $Q$, and let $\mathcal{T}$ denote the space of allowable translations. Then, the *earth mover's distance between $P$ and $Q$ under translation* is defined to be

$$\text{EMD}\,\langle P, Q\rangle \;=\; \min_{t\in\mathcal{T}} \text{EMD}\,(P+t, Q) \;=\; \min_{t\in\mathcal{T}} \min_{M\in\mathcal{M}} \sum_{(p,q)\in M} \|(p+t)-q\|_2\,.$$

Our approach is to first modify Indyk and Thaper's randomized embedding algorithm to a deterministic version. (In contrast to their algorithm, ours is significantly less sensitive to translation). Then, we reduce the problem to point pattern searching with translation under the symmetric difference distance and apply our embedding technique for the point pattern search with translation under symmetric difference.

In Chapter 5 we present the following theorem, which provides an embedding from EMD to symmetric difference, but which does not involve translation. Define $\mathbb{R}^d_\Delta(=n)$ to be the collection of all sets of points of cardinality $n$ over $\mathbb{R}^d$ whose coordinates are over the interval $[0,\Delta)$. **COMMENT:** [Minkyoung, I have added the unit-distance assumption and restated the theorems in this form. I also added that $\Delta = n^{O(1)}$. Please check.] Following Indyk and Thaper, we imagine that the points have been derived from some measurement process, and the actual coordinates of the points are known to some minimal precision, determined by the sensor's limitations. Let us assume that distances have been uniformly scaled so that this minimal

precision is roughly one unit. We assume that the distance between any two points (even after alignment) is defined to be the maximum of 1 and the actual distance. We call this the *unit-distance assumption*.

**Theorem 3** *Given a positive real parameter $\Delta = n^{O(1)}$ and a constant dimension $d$, there exists an embedding function $\Lambda\colon \mathbb{R}^d_\Delta(=n) \to \mathbb{Z}^m$, for $m = O(n\Delta^d \log^2 \Delta)$, such that for any two sets $P, Q \in \mathbb{R}^d_\Delta(=n)$ under the unit-distance assumption*

$$\sqrt{d} \cdot \mathrm{EMD}\,(P,Q) \;\;\leq\;\; |\Lambda P \ominus \Lambda Q| \;\;\leq\;\; 6\sqrt{d}(\log \Delta)\mathrm{EMD}\,(P,Q)\,.$$

We then show that, by combining this result with Theorem 3, the following translation-invariant embedding is obtained.

**Theorem 4** *Given a positive real parameter $\Delta = n^{O(1)}$, a constant dimension $d$, and failure probability $\beta$, there exists a randomized embedding $\Gamma\colon \mathbb{R}^d_\Delta(=n) \to \ell_1^m$, for $m = O(n\Delta^d \log^2(n\Delta) \log \frac{1}{\beta})$, such that for any two sets $P, Q \in \mathbb{R}^d_\Delta(=n)$ under the unit-distance assumption, with probability at least $(1-\beta)$*

$$\frac{\sqrt{d}}{17 \log k}\mathrm{EMD}\,\langle P,Q\rangle \;\;\leq\;\; \|\Gamma P - \Gamma Q\|_1 \;\;\leq\;\; (12\sqrt{d}\log k \log \Delta)\mathrm{EMD}\,\langle P,Q\rangle\,,$$

*where $k = 2n\Delta^d$.*

This embedding results show that point sets in $\mathbb{R}^d_\Delta(=n)$ with translation under EMD can be embedded into a vector space in $\ell_1$. We can apply a general nearest neighbor data structure for finding the closest one to given a query point set.

Complete details of our algorithm and results can be found in Chapter 5.

## 1.2 Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2 a review of the relevant literature in this area is presented. The alignment-based point pattern matching algorithm and its analysis are presented next in Chapter 3. In Chapter 4, the translation-invariant embedding of point sets under symmetric difference is presented. In Chapter 5, the algorithm for embedding point sets under the EMD metric into the symmetric difference distance is presented. Finally, Chapter 6 summarizes the main results of this dissertation, and discusses topics for possible future research.

# Chapter 2

## Literature Review

As mentioned in the introduction, the geometric point pattern matching problem is a fundamental computational problem and has numerous applications in diverse areas. There are many ways in which to formulate point pattern matching as a computational problem. In this section we will describe some formulations that have been considered in recent decades. Depending on the combination of problem characteristics (for example, the choice of distance measure, the space in which the points reside, the group of allowable transformations), the algorithmic approaches vary significantly. Since many point pattern matching problems are formulated as optimization problems (e.g., find the transformation minimizing some distance measure) each formulation can be considered both in the context of computing an optimal solution or an approximation to the optimum. Since the field is quite extensive, we have focused primarily on results from the field of computational geometry that are most relevant to this dissertation.

## 2.1 Exact Point Pattern Matching

Perhaps the simplest version of the point pattern matching problem is the formulation in which the no errors or outliers are tolerated. Given two point sets $P$ and $Q$, the question is whether the two sets are congruent, that is, they are identical after some transformation.

Observe first that the case of translation is trivial since it suffices to align some common reference point, like the centers of mass or centroids, of the two sets. For the case of rigid motion (translation and rotation), Atkinson [8] observed that the optimal transformations must map the centroid $P$ to the centroid of $Q$, after which the problem reduces to computing the optimal rotation. Computing the optimal rotation can be reduced to a string pattern matching problem. He presented an $O(n \log n)$ time algorithm in the plane. First, the centroid $c$ of $P$ is computed and the points are represented in polar coordinates with respect to $c$. These coordinates are sorted in angular order and encoded as a string. The process is repeated for $Q$. To deal with the circular nature of polar coordinates, one of the strings is doubled through concatenation, and the algorithm searches for an occurrence of the other through any standard string-matching algorithm.

Alt, Mehlhorn, Wagener, and Welzl [4] presented a solution to the exact problem in 3d. Again the point sets are translated so their centroids coincide with the origin. Each of the point sets is then projected onto the unit sphere and the convex hull is computed. It is then shown how to determine whether the convex hulls are congruent in $O(n \log n)$ time. In general they show that, for any $d \geq 3$, it is possible

to reduce the $d$-dimensional matching problem to $n$ matching problems in dimension $d-1$. This implies an $O(n^{d-2}\log n)$ time algorithm, for any dimension $\geq 3$.

Sprinzak and Werman [57] consider exact pattern matching under affine transformations. First, $P$ and $Q$ are transformed to their second moment matrices in the group of unit matrices. Then it is shown that $P$ and $Q$ can be matched under an affine transformation if and only if their unit matrices can be matched under rotation. The running time of this algorithm is the same as in the rigid case, since the normalizing transform can be computed in linear time.

## 2.2 Point Pattern Matching for Noisy Data

The case of exact point pattern matching is not very useful in practice because point coordinates are often typically subject to measurement and discretization errors. Therefore, it is important to consider pattern matching in a noisy setting. When noise is present, it is necessary to define a measure of similarity. We consider a number of such measures in the subsequent sections.

## 2.3 One-to-One Matching

We first consider matching functions that are required to be one-to-one. In such cases an injective function is given from one set to the other, and then some function of the resulting distances is computed. The *bottleneck distance* is a well known example. It is defined to be

$$dist(P,Q) \;=\; \min_{f}\min_{p\in P}\|f(p)-p\|,$$

where the minimum is taken over all injective functions $f : P \to Q$.

Alt, Mehlhorn, Wagener, and Welzl [4] presented an $O(n^6)$ time algorithm for the translation that minimizes the bottleneck distance in the plane. Their algorithm operates by parametric search. In particular, they assume that a distance parameter $\delta$ is given, and they apply binary search to obtain the optimal value of $\delta$. (See [46] for a more detailed explanation of parametric search.) They observed that, in any optimal placement, there exist two pairs of points in $P$ and $Q$ such that $\|p_1 - q_1\| = \|p_2 - q_2\| = \delta$. Let $C_1$ and $C_2$ denote the circles centered at two points $p_1, p_2 \in P$, respectively, both with radius $\delta$. Clearly, the corresponding points $q_1, q_2 \in Q$ lie on $C_1$ and $C_2$, respectively. Next, the algorithm computes the trace of another point $q_i \in Q$ as an algebraic curve of constant degree subject to the constraint that $q_1$ and $q_2$ on these circles. They show that the resulting curve can intersect a circle of radius $\delta$ centered at any other point $p \in P$ at most 12 times. Thus, a point $q_i \in Q$ generates at most six intervals on the circle of radius $\delta$ centered at each $p \in P$, and so the maximum number of intervals for each $q_i$ is at most $6n$. $O(n^2)$ intervals are obtained for all $Q$. For each interval at least one of the correspondences is changed. The algorithm checks whether there exists a one-to-one correspondence in the interval by computing a perfect matching on the graph in which each edge represents the fact that two points $p \in P$ and $q \in Q$ are joined by an edge and $dist(p, q) \leq \delta$. Since there are the pairs $(p_1, p_2)$ and $(q_1, q_2)$, all possible pairs are tried. Since there are $O(n^4)$ such pairs, and the perfect matching can be computed in $O(n^2)$ time, the total running time is $O(n^6)$. Efrat and Itai [23] presented a modest improvement to this algorithm by showing how to compute the

perfect matching in $O(n^{1.5} \log n)$ time.

Observe that even for this relatively simple case of planar point pattern matching under translation, the running times are quite high. This has prompted research into more efficient algorithms by resorting to approximation.

Heffernan and Schirra [30] considered an approximate version of the point pattern matching for one-to-one matching under rigid motions. The basic idea of their algorithm is as follows. First, axis-parallel grid lines are drawn on the plane with width and height $\gamma$, and each point of $P$ and $Q$ is snapped to the nearest intersection of two grid lines, called a *grid point*. They compute a one-to-one matching by reduction to max-flow on a customized flow network $G(s, t, P, Q, C)$, where $s$ is connected to all points in $P$, and $t$ is connected to all points in $Q$. To speed-up the processing, a compressed graph is considered. They build a set $C$ satisfying the following conditions. Let $a_i$ denote the grid cell for $p_i$, and let $b_j$ denote the grid cell for $q_j$. If $dist(a_i, b_j) \leq \delta$ then, a vertex $c_{ij}$ is entered in the set $C$ and connected to $p_i$ and $q_j$. In this setting, if the resulting max-flow value is $n$, then there exists a perfect matching on the original graph within distance $\delta = \sqrt{2}\sigma$. Since the grid snapping process introduces an error, this algorithm yields an approximate version for the one-to-one problem. Note that the resolution of the grid determines the approximation error. For a given (optimal) distance parameter $\varepsilon$, they observe that the distance between the centroids of the two point sets is at most $\varepsilon$. With this fact, they solve the translation problem easily with an additional error of $\varepsilon$. For the case of rotation, they pick the farthest point from a fixed center and draw a circle. By subdividing the circular arc with a function of $\gamma$, they compute a candidate set

of rotations in which subdivided arcs have corresponding points within a distance depending on both $\varepsilon$ and $\gamma$. Then, check all possible rotations of the candidate set. Combining the two, they solve the problem under rigid motion approximately in running time $O(n^{2.5}(\varepsilon/\gamma)^5)$.

Utrianinen presented a $(1+\varepsilon)$ approximate algorithm for one-to-one point pattern matching under rigid motions and the least squares error. First, given two point sets, an optimal translation is computed by aligning their centroids. Computing the optimal rotation is a more difficult problem. He shows that it is a special case of the vector-weighted bipartite matching problem, for which no polynomial time algorithm is known. In 2-d, it can be solved approximately by sampling a collection of $\frac{2\pi}{\varepsilon}$ angles uniformly about the circle, testing each, and returning the best solution among them. This yields a $(1+\varepsilon)$ approximate solution. The running time of this algorithm is $O(n^3/\varepsilon)$.

## 2.4 Many-to-One Matching and Hausdorff Distance

As mentioned earlier, in applications where there may be spurious or duplicate points in the pattern set, one-to-one matchings may not be appropriate. When many-to-one matches are allowed, it is possible to match each pattern point against its closest reference point.

A commonly used distance function in this context is the directional Hausdorff distance. Recall that, given two point sets $P$ and $Q$, the *directional Hausdorff*

*distance*, denoted $h(P, Q)$, is defined to be

$$dist(P, Q) = \max_{p \in P} \min_{q \in Q} \|pq\|,$$

where $\|pq\|$ denotes the Euclidean distance between points $p$ and $q$.

Note that the definition is asymmetric since every point of $P$ must be mapped to some of point of $Q$, but not vice versa. Each point of $Q$ can be matched with many points of $P$. The directional Hausdorff distance is a natural choice in applications where the pattern $P$ is expected to match some subset of the reference $Q$. In applications where it is desired that every point of $P$ matches some point of $Q$ and vice versa, a bidirectional similarity measure may be more appropriate, such as the *bidirectional Hausdorff distance*, which is defined to be $\max(h(P, Q), h(Q, P))$ [2, 4, 23, 31, 41]. Unless stated otherwise, we will use the term Hausdorff distance to denote the directional Hausdorff distance.

A number of algorithms have been proposed for computing the optimal pattern matching under the Hausdorff distance [14, 34]. The best-known algorithm for determining the rigid motion that minimizes the directional Hausdorff distance between two planar point sets $P$ and $Q$ of sizes $m$ and $n$, respectively, runs in $O(m^3 n^2 \log^2 mn)$ time [14]. Chew, Goodrich, Huttenlocher, Kedem, Kleinberg, and Kravets compute an optimal solution by first solving a decision problem for a given distance threshold $\delta$, then compute the optimal value of $\delta$ by parametric search [1]. They apply the observation that, if $P$ and $Q$ are an optimal placement, each point of $P$ lies within a disc of radius $\delta$ centered a point of $Q$. For the special case of translation only, discs of radius $\delta$ are drawn centered at each point of $Q$. For each point

$p \in P$, the resulting disc sets are translated by the vector $-\vec{p}$. Thus, $m$ disc sets are generated. In this context, each disc set represents the transformation spaces of each point of $Q$. They show that there is a point that is contained within $m$ discs if and only if the Hausdorff distance between $P$ and $Q$ is at most $\delta$.

To solve the general problem rotation must also be considered. They approach this by treating this as a kinetic algorithm parameterized by the rotation angle $\theta$. Although $\theta$ is a continuous function, point correspondences only changed when the overlap of discs changes. They count the number of the critical events at which two discs are tangent or three circles meet at a point. They present an upper bound of $O(m^3n^2)$ on the number of such events and, by using dynamic Vornoi diagrams, they show that the decision problem can be solved in $O(m^3n^2 \log mn)$ time. Thus, the overall running time of their algorithm is $O(m^3n^2 \log^2 mn)$.

## 2.5 Approximation Algorithms for the Hausdorff Distance

Given the high complexity of the above algorithm, it is natural to consider approximation algorithms. In a traditional approximation formulation, the objective is to output a solution that is within a factor of $(1 + \varepsilon)$ times the optimal distance, for some $\varepsilon > 0$. An alternative formulation, which is often used in point pattern matching, is to return a distance that is within $(1 + \varepsilon)\delta$, where $\delta$ is a user-supplied distance threshold. (This choice motivated by the fact that, in many applications, errors result from digitization to some known resolution, and there is little point in achieving an approximate solution that is of higher resolution than this.) In

addition, random sampling techniques can be applied for further performance improvement. The *approximation ratio* of a pattern matching algorithm is defined to be the maximum ratio, over all input instances, between the distance produced by the algorithm and the optimum distance.

Alt, Aichholzer, and Rote [2] presented a very simple (almost trivial) algorithm to achieve a constant factor approximation to the point pattern matching problem under translation bidirectional Hausdorff distance. The algorithm works by first computing a representative point, called *Steiner point*, of each of the two sets and then computing the translation that aligns these two points. They show that the approximation factor depends only on the *Lipschitz constant* $c$ of the Steiner point. In particular, given two point sets $P$ and $Q$ and an associated distance function dist, the associated Steiner points, $s(P)$ and $s(Q)$, respectively, should satisfy the following two conditions, for any allowed transformation $t$:

$$s(tP) \;=\; t(s(P)) \quad \text{and} \quad \|s(P) - s(Q)\| \;<\; c \cdot \text{Dist}(P, Q).$$

Goodrich, Mitchell, and Orletsky [28] considered the point pattern matching problem under a number of different transformation groups and in various dimensions for the (directional) Hausdorff distance. They presented a simple approximation algorithm based on alignments of small tuples of points. For example, in the case of rigid transformations in the plane, their alignment algorithm proceeds as follows. Let $(p_1, p_2)$ denote a pair of points having the greatest distance in $P$, that is, a *diametrical pair*. For each distinct pair $(q_1, q_2)$ in $Q$, their algorithm computes a rigid transformation matching $(p_1, p_2)$ with $(q_1, q_2)$ as follows. First, it applies to $P$

a translation that maps $p_1$ to $q_1$. Then it performs a rotation about $p_1$ (after translation) that aligns the directed line segment $\overrightarrow{p_1p_2}$ with $\overrightarrow{q_1q_2}$. The rigid transformation $E$ resulting from the composition of this translation and rotation is then applied to the entire pattern set $P$, and the Hausdorff distance $h(E(P), Q)$ is computed. After repeating this for all pairs $(q_1, q_2)$, the transformation with the smallest Hausdorff distance is returned. The running time of this algorithm is $O(n^2 m \log n)$ because, for each of the $n(n-1)$ distinct pairs of $Q$, we compute the aligning transformation $E$ in $O(1)$ time, and then, for each point $p \in P$, we compute the distance from $E(p)$ to its nearest neighbor in $Q$. Nearest neighbors queries in a planar set $Q$ can be answered in time $O(\log n)$ after $O(n \log n)$ preprocessing [22]. They prove that this algorithm returns an aligning transformation whose Hausdorff distance is at most a factor of 4 larger than optimum Hausdorff distance. (In Chapter 3, we will present an algorithm that improves the approximation factor. We show that as the ratio of the diameter and the optimal Hausdorff distance tends to infinity, the approximation factor our algorithm approaches 3.)

Indyk, Motwani, and Venkatasubramanian [39] presented an $\varepsilon$-approximation algorithm for the planar point pattern matching problem under rigid transformations for the Hausdorff distance. That is, the Hausdorff distance produced by their algorithm is at most a factor of $(1 + \varepsilon)$ greater than the optimum, where $\varepsilon > 0$ is a user-supplied parameter which is given an estimate of the optimum Hausdorff distance. Their algorithm first computes the diametrical pair $p_1$, $p_2$ of $P$. Then it enumerates all the pairs $(q_1, q_2)$ of points in $Q$ that lie within a distance of $2\varepsilon$ with respect to $\overline{p_1p_2}$. They show that the number of such pairs is at most

$O(\min(\varepsilon n^{4/3}\Delta^{1/3}\log n, n^2))$, where $\Delta$ is the ratio of the distances between the farthest and closest pairs of points. To obtain the desired accuracy, they employ the approach of Heffanan and Schirra [31] for discretizing transformation space. Let $C_1$ and $C_2$ denote circles centered $p_1$ and $p_2$, respectively, each of radius $\varepsilon$. First, $C_1$ is covered with a grid of width and height with $\gamma$. Then, for each grid point, a circular arc of radius $\overline{q_1 q_2}$ centered at the grid point and intersecting the disc $C_2$ is computed. The arc is subdivided into subarcs of length $\gamma$. Now, $q_1$ and $q_2$ can be placed on the grid points. By using the alignment algorithm, they compute a transformation for each placement and compute the Hausdorff distance of this transformation. The running time is $O(m\cdot\min(\varepsilon n^{4/3}\Delta^{1/3}\log n, n^2)\cdot\min(\varepsilon^2, \log n))$ where $m$ is the number of elements in $P$.

Cardoze and Schulman [11] also gave efficient $\varepsilon$-approximation algorithms for 2-dimensional point pattern matching under both translation and rigid transformation for the Hausdorff distance. Their algorithms are randomized. For the case of translations, their algorithm operates by reduction to a collection of $O(n)$ 1-dimensional point pattern matching problems, each of which is then reduced to computing a convolution. They exploit the fact that convolutions can be computed efficiently in $O(n\log n)$ time using the fast Fourier transform. Its running time is $O(n^2\log n + \log^{O(1)}\Delta)$ for any fixed precision parameter and any fixed success probability.

## 2.6 Robustness and the Partial Hausdorff Distance

One shortcoming of simple Hausdorff-based matching is the lack of sensitivity to outliers, that is, the presence of points in the pattern set that do not match any part of the reference set. Such an outlying point may distort the Hausdorff distance by an arbitrary amount. A distance function is said to be *robust* if it is insensitive to the presence of outliers. One approach to reduce sensitivity to outliers is to allow up to $k$ points to not be matched, where $k$ is a value chosen by the user, varying from a constant up to a constant factor times $n$.

The partial Hausdorff distance [35] handles this isssue well. Given two point sets $P$ and $Q$, the Partial Hausdorff distance is defined by

$$h_K(P,Q) = K^{th}_{p\in P} \min_{q\in Q} \|p-q\|,$$

where $K^{th}_{p\in P}$ denote the K-th ranked value in the set of distance.

Another approach is by a method called the Largest Common Point Set (LCP). Given two point sets $P$ and $Q$, it finds a transformation that maximizes the cardinality of the subset $Q' \subseteq Q$, such that the Hausdorff distance or bottleneck distance is at most $\delta$. Let $\mathrm{LCP}(P,Q)$ denote the size of the optimal solution to the above problem. The degree of robustness can be controlled by two parameters, the distance $\delta$ and the cardinality $|Q'|$. Thus, two kinds of approximate algorithms exist: size approximation and distance approximation. The *$\alpha$-LCP* is a size-approximate version of LCP such that the solution subset $Q'$ is of cardinality at least $\frac{1}{\alpha}\mathrm{LCP}(P,Q)$. The *$\beta$-distance LCP* is a distance-approximate version where the result distance is less than $(1+\beta)\delta$.

In recent work, Choi and Goyal [17] present various combinations of LCP algorithms. First, they give a 4-distance approximation algorithm for LCP problem under the bottleneck distance in 3-d. As in the algorithm of Goodrich *et al.* [28], for each pair $(p_1, p_2) \in P$ and $(q_1, q_2) \in Q$, they align $p_1$ to $q_1$ by translation. Then, align $p_2$ with $q_2$ by rotating at $p_1$. To fix a transformation in 3-d, one more correspondence should be selected. For each point $p \in P - \{p_1, p_2\}$, they compute all possible rotation angles where $p$ lies within distance $\varepsilon$ of $q \in Q$. Since the maximum overlap among all angles represents the maximum matching, this will be the solution of the LCP problem. Since one of combination of all pairs $(p_1, p_2)$ and $(q_1, q_2)$ is the diameter pairs of an optimal solution, this algorithm returns a 4-approximation. In addition, they present an $\alpha$-approximation for the exact LCP ($\delta = 0$) in which the result set is larger than the optimal solution by at most $\frac{1}{\alpha} LCP(P, Q)$. Let $n$ denote the number of elements in $P$. They subdivide $P$ into $\frac{n}{\alpha}$ disjoint subsets, each of size $\alpha$. They observe that, by a pigeonhole argument, at least one of these subsets must have a valid pair in LCP$(P, Q)$. They improve the running time from $O(n^2)$ to $O(\frac{n}{\alpha} \cdot \alpha^2)$.

## 2.7 Point Pattern Similarity Search

In this section, we present a new type of problem that finds similar sets in a large database of point sets, as opposed to the classical problem of comparing a single pair of point sets.

**Definition 2** Let $\mathcal{T}$ be geometric transformation group. Let *dist* be distance func-

tion. Let *Dist* be distance function under the set of transformations $\mathcal{T}$. Given a collection of point sets $\mathcal{P}$, *point pattern similarity search* is defined as finding the point set $P^* \in \mathcal{P}$ that minimizes the distance. That is,

$$P^* = \underset{P \in \mathcal{P}}{\operatorname{argmin}} \; Dist(P, Q) = \underset{P \in \mathcal{P}}{\operatorname{argmin}} \; \min_{t \in \mathcal{T}} \; dist(tP, Q).$$

$\diamondsuit$

The trivial solution would be to compute the distance from the query point set to each point set of the collection (by any single-pair algorithm) and return the closest. However, this brute-force search is not acceptable in cases where the total number of point sets of $\mathcal{P}$ is very large.

As before, several problems can be defined depending on criteria, such as the distance function, error model, and permitted transformations. Note that these types of problems have not been heavily studied from the perspective of computational geometry.

First, let us consider perhaps the simplest problem of determining whether there is a congruent point set from a set of point sets under translation. One idea would be adapt the approach of Atkinson [8]. Recall that his algorithm reduces the point-set congruence problem to a string search problem. If successful, the point pattern matching problem would then be reduced to a dictionary search. There are some technical difficulties to be faced in a straightforward implementation of his algorithm. For example, Atkinson's approach of storing points in polar coordinates would result in real-valued (irrational) quantities, which would then need to be encoded as strings. If rotation is not involved and the input coordinates are integers,

however, it is possible to encode point coordinates in a manner that is translation invariant, for example, relative to the lower left corner of the point set's bounding box.

Another well-known algorithm for point pattern similarity search is *geometric hashing* [61], which is a framework developed for matching geometric features in a database of point sets. The intuition behind this method is straightforward. For each point set, it generates all possible transformed sets and generates a data-structure like a hash table. When a query point set is given, it evaluates whether there exists a point set matching with all points in the query point set. This approach works well when the number of point sets are relatively small. Otherwise, the performance may be degraded if there exist many candidate sets that are returned for each point in the query point set. Note that the space complexity and query times grow exponentially with the number of degrees of freedom in the group of transformations.

When outliers exists, the point pattern similarity search problem under geometric transformation becomes significantly more challenging. The presence of outliers renders most methods based on finding a small number of canonical alignments invalid or requires the consideration of a large number of possible alignments. This has an adverse effect either on space or on query time. In some cases, even computing individual distances is a nontrivial task. One approach to circumventing this problem is based on the notion of embedding, which we discuss next.

### 2.7.1 Point Pattern Similarity Search and Embeddings

In point pattern similarity searching, we are given a database of point sets, which are to be preprocessed, so that, given a query point set, it is possible to find the closest matches from the database efficiently subject to a set of allowable aligning transformations. An important tool in this area is the notion of an *embedding*, which maps points from one metric space to another. Through embeddings, we can map a proximity search problem in a complex metric space, to an equivalent problem in a simpler metric space. In this section we introduce a few of the most relevant concepts from this field. Recall the following definitions.

**Definition 3** Given metric spaces $(X, d)$ and $(X', d')$ a map $f : X \to X'$ is called an *embedding*. An embedding is *distance-preserving* or *isometric* if for all $x, y \in X$,

$$d(x, y) = d'(f(x), f(y)).$$

The *contraction* of $f$ is the maximum factor by which distances are shrunk, i.e.,

$$\max_{x,y \in X} \frac{d(x, y)}{d'(f(x), f(y))},$$

the *expansion* or *stretch* of $f$ is the maximum factor by which distances are stretched:

$$\max_{x,y \in X} \frac{d'(f(x), f(y))}{d(x, y)},$$

and the *distortion* of $f$ is the product of the contraction and expansion. $\diamondsuit$

Thus, one approach to solving point pattern similarity search problem is to design a low-distortion, transformation-invariant embedding which maps each point set into a well-known metric space. Then a nearest neighbor search algorithm can

be applied on the embedded point sets of the database. I present some literature related to this approach.

Let us begin with the case of point pattern similarity search with translation under symmetric difference. A popular method in computer vision to determine similarity of point patterns is through the use of invariant features, that is, some statistic of the set that is invariant under some given set of transformations [13, 29, 33, 44].

An interesting example of an invariant for point sets is based on the observation that the distance between a pair of points is invariant under rigid transformations. The *distance histogram* of a set of $n$ points is the set of $O(n^2)$ inter-point distances defined by all pairs of points in the set.

If two point sets are similar, then clearly their distance histograms are similar as well. Note that histograms can be compared using the $\ell_1$ or EMD metrics, and so standard nearest neighbor search algorithms can be applied. However, the converse of this observation does not hold. For example, consider the following two sets:

$$P = \{0, 1, 4, 10, 12, 17\} \qquad \text{and} \qquad Q = \{0, 1, 8, 11, 13, 17\}.$$

These sets have identical distance histograms but only three points out of six are matched. Thus, the distortion of the resulting embedding is unbounded. Point sets that have equal distance histograms are called *homometric*. Homometric sets and their properties have been studied in mathematics and computational geometry [21, 50, 56]. (In Chapter 4, we present a theoretical bound for point pattern similarity searching under translation with outliers based on the symmetric differ-

ence distance.)

A common method for comparing histograms, and distributions in general, is the Earth-Mover's Distance (EMD). The *EMD distance* is defined as a deformation distance between two sets of equal cardinality, that is, the total distance needed to move each point of one set to a corresponding point of the other set. (See Chapters 1 and 5 for definitions.)

Indyk and Thaper [40] consider an image database search problem under the earth mover's distance. They present a randomized embedding of the EMD distance into $\ell_1^k$ (real $k$ dimensional space under the $\ell_1$ metic). The expected distortion of their embedding is $O(\log \Delta)$, where $\Delta$ is the diameter of both point sets.

Here is a sketch of their method. Let $P$ and $Q$ denote two point sets in $\mathbb{R}^d$ each of cardinality $n$. Let $V = P \cup Q$. Assume that the smallest interpoint distance is 1. Assume first that $\Delta$ is rounded up to the next larger power of 2, and apply a random shift to the coordinate axes. Generate a series of grids, with side lengths $2^i$, for $i \in \{-1, 0, 1, 2, \ldots, \log \Delta\}$. Let $G_i$ denote grid of side $2^i$. For each $G_i$, compute a vector $v_i(P)$ with dimension $(\Delta/2^i)^d$ such that each dimension represents the number of elements on in cell of the grid. They show that there is a constant $c$, such that the upper bound on the EMD distance between $P$ and $Q$ is at most $c \sum_{i=1}^{\log \Delta} \|v_i(P) - v_i(Q)\|_1$. For the lower bound, they show that the EMD distance is at least $\Omega(\log \Delta)$ in expectation. Note that their formulation does not allow for any transformation between the two sets.

Shirdhonkar and Jacobs [55] presented a wavelet-based variant of EMD. They show that their measure is a metric, and that the ratio of the EMD to the wavelet

EMD is bounded within some constant. This wavelet EMD metric is based on the weighted wavelet coefficients of the difference of two histograms and the computation requires only linear time, so that they improve the computation time of EMD with small constant distortion. Also, empirically they showed that their algorithm is more accurate than that of Indyk and Thaper.

Farach-Colton and Indyk [24] presented an embedding algorithm for point sets with translation under the Hausdorff metric. Letting $s$ denote the cardinality of the database, and $d' = s^2(1/\epsilon)^{O(d)}$, they show that their embedding achieves a distortion of $(1 + \epsilon)$. The embedded space has size that is quadratic of the size of database, which is too large for most practical applications.

# Chapter 3

# Improved Approximation Bounds for Planar Point Pattern Matching

## 3.1 Introduction

In this chapter we will consider point sets in the Euclidean plane under rigid transformations (translation and rotation). Distances between two point sets $P$ and $Q$ will be measured by the *directed Hausdorff distance*, denoted $h(P, Q)$, which is defined to be

$$h(P, Q) = \max_{p \in P} \min_{q \in Q} \|pq\|,$$

where $\|pq\|$ denotes the Euclidean distance between points $p$ and $q$.

The directional Hausdorff distance is natural in applications where the pattern $P$ is expected to match some subset of the background $Q$. In applications where it is desired that every point of $P$ matches some point of $Q$ and vice versa, a bidirectional similarity measure may be more appropriate such as the *bidirectional Hausdorff distance*, which is defined to be $\max(h(P, Q), h(Q, P))$. When matching with bidirectional similarity measures it may be possible to identify a global reference point (such as the centroid of each set) about which to anchor the alignment.

This is not possible, however, under the directed Hausdorff distance. Throughout, unless otherwise specified, we use the term *Hausdorff distance* to denote the directed Hausdorff distance.

A simple and natural approach is to consider transformations induced by aligning a small subset of points from one set to the other. This is arguably the simplest and most easily implemented algorithm for approximate pattern matching, and it is the basis of some of the most popular methods in computer vision such as RANSAC [26]. Goodrich, Mitchell, and Orletsky [28] were the first to prove an upper bound on the approximation ratio of such a simple alignment-based algorithm. They considered point pattern matching under a number of different transformation spaces and in different dimensions. For the case of rigid transformations in the plane, their algorithm computes a diametrical pair for $P$ and then computes for every pair of distinct points of $Q$ a rigid transformation that aligns these pairs. (Details are presented in the next section.) It returns the transformation achieving the minimum Hausdorff distance. It runs in $O(n^2 m \log n)$ time. They prove that it returns an aligning transformation whose Hausdorff distance is at most a factor of 4 larger than the optimum Hausdorff distance.

In this chapter, we revisit the approximation ratio for the alignment-based algorithm of Goodrich, Mitchell, and Orletsky [28]. Their algorithm is based on aligning points one by one, and henceforth we refer to this as *serial alignment*. (We will describe their algorithm detail Section 3.2.) Given the simplicity of this approach, it is natural to establish good bounds on its performance. This also points toward larger questions such as what are the best approaches to alignment-

based matching and what are the inherent limits on the accuracy of alignment-based matching. We show that it is possible to improve on their approximation ratio of 4. Our approach has the same running time as theirs and, like theirs, is very easy to implement. It is based on a minor modification that selects the transformation that best aligns the entire subset of points, which we call *symmetric alignment*. Let $A_{ser}$ and $A_{sym}$ denote the approximation ratios for these respective algorithms.

Rather than just considering the worst-case approximation ratios, we analyze the approximation ratios of these algorithms in a manner that is sensitive to the optimal Hausdorff distance. For each problem instance $P$ and $Q$, we define an geometric parameter $\rho$, called the *distance ratio*, to be half the ratio of the diameter of $P$ to the optimum Hausdorff distance between $P$ and $Q$. (A formal definition is given in Section 3.3.) We show that as the distance ratio increases, the accuracy of the approximation increases as well. We feel that this analysis is useful because large values of $\rho$ often arise in applications. For example, in document analysis and satellite image analysis, the ratio of the diameter of a typical pattern ranges from tens to hundreds of pixels, while the expected digitization error is on the order of a single pixel. Let $A_{ser}(\rho)$ and $A_{sym}(\rho)$ denote the approximation ratios for serial and symmetric alignment, respectively, as a function $\rho$. Our results are summarized in Table 3.1. The quantity $c_\infty$ is the approximation ratio $A_{ser}(\rho)$ in the limit as $\rho$ approaches infinity, which we show later to be roughly 3.19.

Our results imply that the approximation ratio for symmetric alignment is better than that of serial alignment for almost all but very small values of $\rho$. Further, for (typical) applications where distance ratio is large, the approximation factor of

Table 3.1: Summary of results for alignment-based approximation. Bounds on the approximation ratios for symmetric and serial alignment algorithms are given as a function of the distance ratio $\rho$, where $c_\infty \approx 3.19$.

| Algorithm | Approximation Ratio | | |
|---|---|---|---|
| | Upper Bound (all $\rho$) | Upper Bound | Lower Bound |
| Symmetric Alignment | $A_{sym} \leq 3.14$ | $A_{sym}(\rho) \leq 3 + \frac{1}{\sqrt{3}\,\rho}$ | $A_{sym}(\rho) \geq 3 + \frac{1}{10\rho^2}$ |
| Serial Alignment | $A_{ser} \leq 3.44$ | $A_{ser}(\rho) \leq c_\infty + \frac{9}{4\rho}$ | $A_{ser}(\rho) \geq c_\infty + \frac{1}{27\rho^2}$ |

symmetric alignment is close to 3. Our results can also be applied to provide a modest improvement in the running time of the $\varepsilon$-approximation algorithm of Indyk *et al.* [39]. Their algorithm uses the simple alignment algorithm as a subroutine. The running time of their algorithm has a cubic dependence on the (upper bound on the) approximation ratio of the alignment algorithm. So, improving the approximation ratio bound by a factor of $f$ results in factor of $f^3$ reduction in the running time of their algorithm.

The remainder of the chapter is organized as follows. In Section 3.2 we present the serial and symmetric alignment algorithms, and we provide an overview of their approximation ratios. In Sections 3.3 and 3.4, we present our upper and lower bounds, respectively, on the approximation ratio for symmetric alignment. In Sections 3.5 and 3.6, we present upper and lower bounds for serial alignment. Finally,

in Section 3.7 we summarize and present concluding remarks.

## 3.2 The Serial and Symmetric Alignment Algorithms

In this section we present a description of the serial alignment algorithm of Goodrich, Mitchell, and Orletsky [28] and review its approximation ratio. We also described our modification of this algorithm, called symmetric alignment. Recall that we are given two planer point sets, a pattern set $P = \{p_1, \ldots, p_m\}$ and a background set $Q = \{q_1, \ldots, q_n\}$. We seek a rigid transformation $E$ that minimizes the (directed) Hausdorff distance from $E(P)$ to $Q$.

**Lemma 3.2.1** (Goodrich, Mitchell, and Orletsky) *The approximation ratio for serial alignment algorithm satisfies* $A_{ser} \leq 4$.

**Proof**: It is easy to see that the serial alignment algorithm is invariant to the initial placement of $P$ in the sense that if $P$ and $P'$ are equal up to a rigid transformation, then (assuming general position) the algorithm will map both these point sets to the same final positions with respect to $Q$ and with the same Hausdorff distance. Thus to simplify matters, we may assume that $P$ has been presented to the algorithm in its optimal position with respect to $Q$. Thus, for each $p_i \in P$, if we let $q_i$ denote its closest point of $Q$ then $\|p_i q_i\| \leq h_{opt}$. Now, run the serial alignment algorithm. We will bound the maximum distance by which any point of $P$ has been displaced relative to its original optimal position, and this will provide the approximation ratio.

First, consider the effect of translation. Since $p_1$ is within distance $h_{opt}$ of $q_1$, each point of $P$ is translated by a distance of at most $h_{opt}$, and so is now within distance $h_{opt} + h_{opt} = 2h_{opt}$ of its closest point of $Q$. Next, consider the effect of rotation. After translation we have $\|p_2q_2\| \leq 2h_{opt}$, from which it follows that the rotation displaces $p_2$ by an additional distance of at most $2h_{opt}$. Since $p_2$ is the farthest point of $P$ from $p_1$, every other point of $P$ is displaced through rotation by at most this distance. We see that after translation and rotation each point of $P$ is at distance at most $2h_{opt} + 2h_{opt} = 4h_{opt}$ from its closest point of $Q$. Therefore the approximation bound is at most $4h_{opt}/h_{opt} = 4$.

(Observe further that after running the algorithm $p_1$ coincides with $q_1$. Also, since rotation can only decrease the distance between $p_2$ and $q_2$, after serial alignment has completed the distance from $p_2$ to $q_2$ is at most $2h_{opt}$. These observations will be used below in Lemma 3.2.3.) □

There are two obvious shortcomings with this algorithm and its analysis. The first is that the algorithm does not optimally align the pair $(p_1, p_2)$ with the pair $(q_1, q_2)$ with respect to Hausdorff distance. One would expect such an optimal alignment to provide a better approximation ratio. This observation is the basis of our algorithm. The second shortcoming is that the use of the triangle inequality in summing the two displacements neglects the presence of any geometrical dependence between these two vector quantities.

We present a new approach called *symmetric alignment*. The algorithm differs only in how the aligning transformation is computed. Let the pairs $(p_1, p_2)$ in $P$ and $(q_1, q_2)$ in $Q$ be chosen in the same way as they are in the serial alignment

algorithm. Let $m_p$ and $m_q$ denote the respective midpoints of line segments $\overline{p_1p_2}$ and $\overline{q_1q_2}$. First, translate $P$ to map $m_p$ to $m_q$ and then rotate $P$ about $m_p$ to align the directed segments $\overrightarrow{p_1p_2}$ with $\overrightarrow{q_1q_2}$. Thus, the only difference is that we align and rotate around the midpoints. It is easy to see that this alignment transformation minimizes the Hausdorff distance between the pairs $(p_1, p_2)$ and $(q_1, q_2)$.

There is a pathological setting in which symmetric alignment can perform arbitrarily poorly compared to the optimum. This happens when, in the optimal Hausdorff placement of $P$, both $p_1$ and $p_2$ share the same closest point of $Q$. The problem is that symmetric alignment assumes that $q_1$ and $q_2$ are distinct (for otherwise the rotation angle is undefined). For this to occur the distance between $p_1$ and $p_2$ (that is, the diameter of $P$) must be less than the distance between the closest pair of points of $Q$. In such a case, the optimal placement results by simply aligning the center of the smallest enclosing disk for $P$ with any point of $Q$. By standard methods in computational geometry (see [22]) this situation an be detected and handled in $O(m + n \log n)$ time, and so henceforth we will assume that it is not an issue.

Before giving our detailed analysis of the approximation ratio we establish a crude bound on the approximation ratio for symmetric alignment, which justifies the benefit of this approach over serial alignment.

**Lemma 3.2.2** $A_{sym} \leq 2 + \sqrt{3} \approx 3.732$.

**Proof**: As in the proof of Lemma 3.2.1, let us assume that $P$ has been presented to the algorithm in its optimal position with respect to $Q$. Thus, each point of $P$

is within distance $h_{opt}$ of its closest point of $Q$. Recall that $q_1$ and $q_2$ denote the closest points of $Q$ to $p_1$ and $p_2$, respectively. For $i \in \{1, 2\}$ let $\vec{v}_i$ denote the vector $\overrightarrow{p_i q_i}$.

First, let us consider translation. To match the midpoints $m_p$ and $m_q$, we translate the pattern set $P$ by the vector $\vec{t} = (\vec{v}_1 + \vec{v}_2)/2$. Since both of these points are within distance $h_{opt}$ of their closest point of $Q$, we have $\|\vec{t}\| \leq h_{opt}$. Next we consider rotation. The distance between the translated point $p_1 + \vec{t}$ and $q_1$ is

$$\left\| \vec{v}_1 - \frac{\vec{v}_1 + \vec{v}_2}{2} \right\| \;=\; \left\| \frac{\vec{v}_1 - \vec{v}_2}{2} \right\| \;\leq\; h_{opt}.$$

Because $(p_1, p_2)$ is a diametrical pair, it follows that all the points of $P$ lie in a lune defined by the intersection of two discs both of radius $\|p_1 p_2\|$ centered at each of these points. Thus, no point of $P$ is farther from the center of rotation ($m_p$) than the apex of this lune, which is at distance $(\sqrt{3}/2)\|p_1 p_2\|$. Since this rotation moves $p_1$ by a distance of at most $h_{opt}$, and $p_1$ is within distance $(1/2)\|p_1 p_2\|$ of $m_p$, it follows that every point $p \in P$ is moved through rotation by an additional distance of at most $\sqrt{3} h_{opt}$.

Thus, we see that each point of $P$ started within distance $h_{opt}$ of its closest point of $Q$, and it has been displaced through translation and rotation by distances of at most $h_{opt}$ and $\sqrt{3} h_{opt}$, respectively. Therefore, after symmetric alignment each point of $P$ is within distance

$$h_{opt} + h_{opt} + \sqrt{3} h_{opt} \;=\; \left(2 + \sqrt{3}\right) h_{opt} \;\approx\; 3.732 \cdot h_{opt},$$

of its closest point of $Q$. The approximation ratio follows immediately.

(Further observe that the points $p_1$ and $p_2$ started out within distance $h_{opt}$ from their closest points of $Q$, and after alignment this is still true. This fact will be used below in Lemma 3.2.3.) □

Even though this crude approximation ratio is already an improvement, it suffers from the same problem as the earlier analysis in that it does not consider the geometric relationship between the translation and rotation vectors. Note that this does not imply that symmetric alignment is better than serial alignment for all input instances. In Fig. 1 we show two informal examples. In the first case serial alignment is better and in the second symmetric alignment is better. The following lemma shows, nonetheless, that the two methods achieve comparable approximation ratios on all input instances.

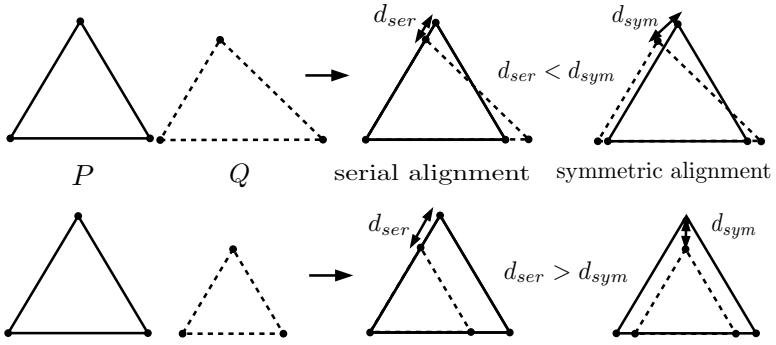


Fig. 1: Comparison of the two algorithms.

**Lemma 3.2.3** *Given two planar point sets $P$ and $Q$, let $d_{sym}$ and $d_{ser}$ denote the Hausdorff distances between $P$ and $Q$ after running symmetric alignment and serial alignment, respectively. Then*

$$d_{sym} \leq d_{ser} + \min\left(\frac{d_{ser}}{2}, h_{opt}\right) \leq \frac{3}{2} d_{ser} \qquad \textit{and}$$

$$d_{ser} \leq d_{sym} + h_{opt} \leq 2 d_{sym}.$$

**Proof**: Let us assume that $P$ has been presented to the symmetric alignment algorithm in the position output by the serial alignment algorithm. Recall that $(p_1, p_2)$ is a diametrical pair from $P$, and $(q_1, q_2)$ is the corresponding pair from $Q$. From the comments made at the end of the proof of Lemma 3.2.1 we know that $p_1$ and $q_1$ coincide, and the distance between $p_2$ and $q_2$ is at most $2h_{opt}$. Clearly, the distance between $p_2$ and $q_2$ is also at most $d_{ser}$. Now, we apply symmetric alignment. In order to align the midpoints of $\overline{p_1p_2}$ with $\overline{q_1q_2}$, each point of $P$ is translated by the distance $\|p_2q_2\|/2$, which is at most $\min(d_{ser}, 2h_{opt})/2$. No rotation is needed. Therefore, we have

$$d_{sym} \;\leq\; d_{ser} + \frac{\|p_2q_2\|}{2} \;\leq\; d_{ser} + \min\left(\frac{d_{ser}}{2}, h_{opt}\right) \;\leq\; \frac{3}{2}d_{ser}.$$

To prove the other bound, let us assume that the point set $P$ has been presented to the serial alignment algorithm in the positions output by the symmetric alignment algorithm. As observed at the end of the proof of Lemma 3.2.2, $\|p_1q_1\| \leq h_{opt}$. If we apply serial alignment to these point sets, each point of $P$ will be translated by at most $\|p_1q_1\|$. No rotation is needed. Therefore,

$$d_{ser} \;\leq\; d_{sym} + \|p_1q_1\| \;\leq\; d_{sym} + h_{opt} \;\leq\; 2d_{sym}.$$

□

## 3.3 Symmetric Alignment: Upper Bound

In this section we derive an upper bound on the approximation ratio for symmetric alignment. As before let $h_{opt}$ denote the optimum Hausdorff distance between $P$ and $Q$ achievable under any rigid transformation of $P$. As mentioned above, our analysis is sensitive to a geometric parameter that is proportional to the ratio of $P$'s diameter to the optimum Hausdorff distance. Define the *distance ratio* to be

$$\rho \;=\; \frac{diam(P)}{2h_{opt}},$$

where $diam(P)$ denotes diameter of $P$. Clearly $h_{opt} \leq diam(P)$ and therefore $\rho \geq \frac{1}{2}$. Recall that $A_{sym}(\rho)$ denotes the approximation ratio for symmetric alignment as a function of the distance ratio. The main result of this section is presented next.

**Theorem 5** *Consider two planar point sets $P$ and $Q$, and let $\rho$ be the distance ratio for $P$ and $Q$. Then*

$$A_{sym}(\rho) \;\leq\; \min\left(3 + \frac{1}{\sqrt{3}\,\rho}, \sqrt{4\rho^2 + 2\rho + 1}\right),$$

*and further $A_{sym}(\rho) \leq 3.14$ for all $\rho$.*

Recall that the distance ratio is large in many applications. This theorem shows that as $\rho$ tends to infinity, the approximation ratio is at most 3. The remainder of this section is devoted to proving this theorem. As usual, it will simplify the presentation to assume that $P$ has been transformed to its optimal placement with respect to $Q$, and we will bound the amount of additional displacement of each point of $P$ relative to this placement. Without loss of generality we may scale space

uniformly so that the optimum Hausdorff distance $h_{opt}$ is 1. Recall that $(p_1, p_2)$ is a diametrical pair from $P$, and $(q_1, q_2)$ is the corresponding pair from $Q$. From our scaling it follows that for $i \in \{1, 2\}$, $q_i$ lies within a disc of radius 1 centered at $p_i$. (See Fig. 2.) Let $m_p$ and $m_q$ denote the respective midpoints of the segments $\overline{p_1p_2}$ and $\overline{q_1q_2}$. Note that $\rho$ is just the distance from $m_p$ to either $p_1$ or $p_2$. Let $\alpha \geq 0$ denote the absolute acute angle between the lines supporting the segments $\overline{p_1q_1}$ and $\overline{p_2q_2}$. If $\rho > 1$, then the two unit discs do not intersect. Hence, after aligning the midpoints by translation, the distance from $p_1$ to $q_1$ is still at most $h_{opt}(= 1)$ (see Lemma. 3.2.2) and the orthogonal distance from $p_1$ to the line $\overline{q_1q_2}$, which is $\rho \sin \alpha$, is less than or equal to the distance from $p_1$ to any point on the line. It follows therefore that $0 \leq \rho \sin \alpha \leq 1$.

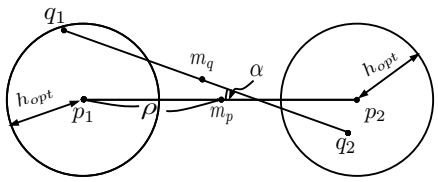


Fig. 2: The positions of the point sets prior to running the algorithm.

Before presenting the details, we give a brief overview of our proof structure. We establish two upper bounds on the approximation ratio, each a function of $\rho$. One bound is better for small $\rho$ and the other better for large $\rho$. We shall show by numerical computations that the two functions cross over one another at a value $\rho^*_{sym} \approx 1.26$ and that $A_{sym}(\rho^*_{sym}) \approx 3.14$. Furthermore, both these functions decrease monotonically as $\rho$ moves away from $\rho^*_{sym}$. We begin by establishing the approximation ratio for low distance ratios.

**Lemma 3.3.1** *For all $\rho$, $A_{sym}(\rho) \leq \sqrt{4\rho^2 + 2\rho + 1}$.*

**Proof**: Let $(p_1, p_2)$ and $(q_1, q_2)$ be as defined in the algorithm. As mentioned in the proof of Lemma 3.2.2, every point of $P$ lies within a lune formed by the intersection of two discs of radius $2\rho$ centered at $p_1$ and $p_2$. Hence, every point of $P$ is within distance $\sqrt{3}\rho$ of the midpoint $m_p$. After applying symmetric alignment, $\overline{p_1p_2}$ and $\overline{q_1q_2}$ are collinear and, for $i \in \{1, 2\}$, the distance $\|p_iq_i\|$ is at most $h_{opt}(= 1)$. Since, $\|p_im_p\| = \rho$, it follows directly that every point of $P$ is within distance $\sqrt{(\rho + 1)^2 + (\sqrt{3}\rho)^2}$ of either $q_1$ or $q_2$. Simplifying yields the desired bound. □

The above approximation ratio is very crude as it considers no points of $Q$ other than $q_1$ and $q_2$. Hence, it is only useful for very small values of $\rho$. To deal with the case of high $\rho$ values we consider the effects of translation and rotation separately. We will characterize the set of possible translations that align $m_p$ with $m_q$ as a function of $\rho$ and the rotation angle $\alpha$. The distance by which an arbitrary point of $p \in P \setminus \{p_1, p_2\}$ can be displaced by the rotation is not only a function of these two parameters, but is also a function of the distance of this point from the center of rotation $m_p$. We will then bound the sum of these two displacement distances. These translation and rotation analyses are presented in Sections 3.3.1 and 3.3.2, respectively, and these results are combined in Section 3.3.3.

### 3.3.1 Translational Displacement

We begin by considering the space of possible translations that align $m_p$ with $m_q$. It will make matters a bit simpler to think of translating $Q$ to align $m_q$ with $m_p$, but

of course any bounds on the distance of translation will apply to case of translating $P$. The translation is given by a vector which we denote by $\vec{t}$. The following lemma bounds the length of this translation vector as a function of $\rho$ and $\alpha$.

**Lemma 3.3.2** *The symmetric alignment's translation transformation displaces any point of $P$ by a vector $\vec{t}$ of length*

$$\|\vec{t}\| \leq \sqrt{1 - \rho^2 \sin^2 \alpha}.$$

**Proof**: Recall that $(p_1, p_2)$ and $(q_1, q_2)$ denote the respective point pairs from the sets $P$ and $Q$ and that $\alpha$ is the acute angle between the lines $\overline{p_1p_2}$ and $\overline{q_1q_2}$. Since we have assumed that $P$ is placed in the optimal position, $q_1$ and $q_2$ must lie inside the circles of radius $h_{opt}(= 1)$ centered at $p_1$ and $p_2$, respectively. For the sake of illustration let us assume that $\overrightarrow{p_1p_2}$ is directed horizontally from left to right and that $\overline{q_1q_2}$ has a nonpositive slope. (See Fig. 3(a).)

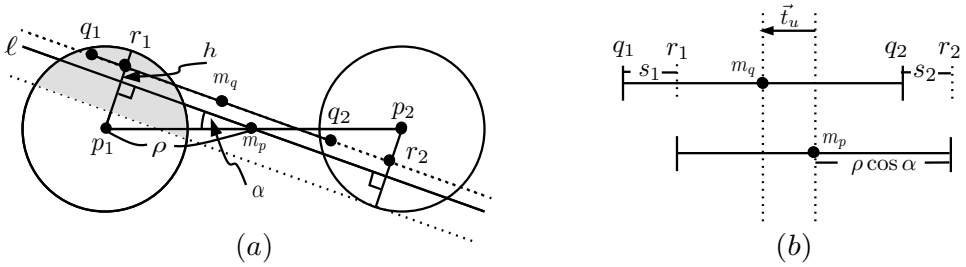


Fig. 3: Analysis of the midpoint translation.

Consider a line passing through $m_p$ that is parallel to $\overline{q_1q_2}$. Let $r_1$ and $r_2$ be the respective orthogonal projections of $p_1$ and $p_2$ onto the line $\overline{q_1q_2}$, and let $s_1$ and $s_2$ denote the respective signed displacements $\|\overrightarrow{r_1q_1}\|$ and $\|\overrightarrow{r_2q_2}\|$ along this line (See Fig. 3(b).) Consider a coordinate system centered at $m_p$ whose positive $u$-axis is

located along a line $\ell$ that is directed from left to right, and whose positive $v$-axis is perpendicular to this and directed upwards. In this coordinate system we have $m_p = (0,0)$, $q_1 = (s_1 - \rho\cos\alpha, h)$, and $q_2 = (s_2 + \rho\cos\alpha, h)$. Thus, $m_q = ((s_1+s_2)/2, h)$, and the translation vector $\vec{t}$ is $m_q - m_p = m_q$. By straightforward calculations we have $|s_1| \leq \sqrt{1-(\rho\sin\alpha + h)^2}$ and $|s_2| \leq \sqrt{(1-(\rho\sin\alpha - h)^2}$. Therefore,

$$\begin{aligned}
\|\vec{t}\|^2 &= u^2 + v^2 = \left(\frac{s_1+s_2}{2}\right)^2 + h^2 = \frac{1}{2}(s_1^2+s_2^2) - \frac{1}{4}(s_1-s_2)^2 + h^2 \\
&\leq \frac{1}{2}(s_1^2+s_2^2) + h^2 \leq \frac{1}{2}\left[(1-(\rho\sin\alpha+h)^2) + (1-(\rho\sin\alpha - h)^2)\right] + h^2 \\
&= 1 - \rho^2\sin^2\alpha.
\end{aligned}$$

Observe that the translation is maximized when $s_1 = s_2$, and this implies that the line $\overline{q_1q_2}$ passes through $m_p$. (This will be used in our lower bound construction in Section 3.4.) □

Given $\alpha$ and $\rho$, let us refer to the set of all valid $(u,v)$ translation vectors as the *translation space*, denoted $\mathcal{T}_\rho(\alpha)$. A more detailed analysis of the set of valid $(u,v)$ pairs shows that the set of possible translational displacements lies within a region of the $(u,v)$ plane as illustrated in the shaded region of Fig. 4, but we will not need this for our subsequent analysis.

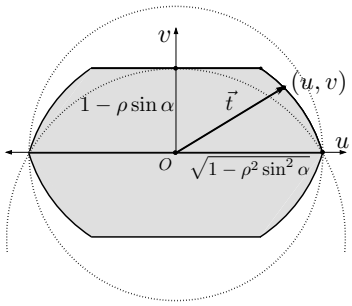


Fig. 4: Translation space, $\mathcal{T}_\rho(\alpha)$, for symmetric alignment.

### 3.3.2 Rotational Displacement

Next we consider the effect of rotation on the approximation error. Unlike translation we consider the placement of points in $P$ because the distance by which a point is displaced by rotation is determined by the distance to the center of rotation. For each point $p \in P$, let $\vec{r}_p$ denote its displacement due to rotation. As mentioned in the proof of Lemma 3.2.2 every point of $P$ lies within a lune formed by the intersection of two discs of radius $2\rho$ centered at $p_1$ and $p_2$. (See Fig. 5(a).) The following lemma describes the possible displacements of a point of $P$ under the rotational part of the aligning transformation. This is done relative to a coordinate system centered at $m_p$, whose $x$-axis is directed towards $p_2$.

**Lemma 3.3.3** *The symmetric alignment's rotation transformation displaces any point $p \in P$ by a vector $\vec{r}_p$ of length*

$$\|\vec{r}_p\| \leq 2\sqrt{3}\rho \sin\frac{\alpha}{2}.$$

**Proof**: Let $\theta$ be the signed angle from $\overrightarrow{m_p p_2}$ to $\overrightarrow{m_p p}$. We may assume $-\pi \leq \theta \leq \pi$. Let $p'$ denote the point after rotating $p$ clockwise about $m_p$ by angle $\alpha$. (See Fig. 5(a).)

By simple trigonometry and the observation that $\triangle m_p p p'$ is isosceles, it follows that the displacement vector $\overrightarrow{pp'}$, length $\|\vec{r}_p\| = \|m_p p\| \cdot 2\sin\frac{\alpha}{2}$. The farthest point of the lune from $m_p$ is easily seen to be the apex, which is at distance $\sqrt{3}\rho$. Thus, $\|\vec{r}_p\|$ is at most $2\sqrt{3}\rho \sin\frac{\alpha}{2}$.

□

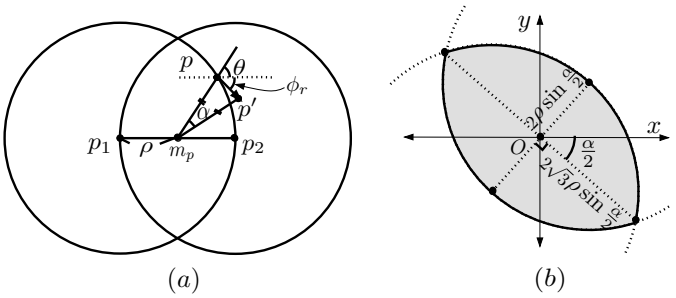


Fig. 5: Analysis of the rotational displacements for symmetric alignment.

As we did for translation, for a given pair of parameter values $\alpha$ and $\theta$ we can define the set of possible vector displacements for all points $p \in P$, which we call the *rotation space*, denoted $\mathcal{R}_\rho(\alpha)$. This is illustrated in Fig. 5(b), but is not needed for the rest of our analysis. Henceforth, when $p$ is clear for context, we will refer to $\vec{r}_p$ as $\vec{r}$.

### 3.3.3 Combining Translation and Rotation

We are now ready to derive the approximation ratio for the symmetric alignment algorithm by combining the bounds on the translational and rotational displacements. Recall that we apply Lemma 3.3.1 to bound $A_{sym}(\rho)$ for small values of $\rho$, and so here we will assume that $\rho \geq \rho^*_{sym}$.

Recall that at the start of the algorithm, the points are assumed to be placed in the optimal positions and that space has been scaled so that $h_{opt} = 1$. Let us fix any point $p \in P$ and let $\vec{t}$ denote its displacement due to translation and let $\vec{r} = \vec{r}_p$ denote its displacement due to rotation. It follows that $p$ has been displaced from its initial position by a distance of $\|\vec{t}+\vec{r}\|$. Since its initial position was within distance $h_{opt}$ $(= 1)$ of some point of $Q$, it is now within distance $\|\vec{t}+\vec{r}\|+1$ of some point of

$Q$. Recalling that $\mathcal{T}_\rho(\alpha)$ and $\mathcal{R}_\rho(\alpha)$ denote the space of possible translational and rotational displacement vectors and that $0 \leq \sin\alpha \leq 1/\rho$, we have

$$A_{sym}(\rho) \;\leq\; \max_{\alpha} \left( \max_{\substack{\vec{t}\in\mathcal{T}_\rho(\alpha) \\ \vec{r}\in\mathcal{R}_\rho(\alpha)}} \|\vec{t}+\vec{r}\| \right) + 1.$$

Unfortunately, determining this length bound exactly would involve solving a relatively high order equation involving trigonometric functions. Instead, we will simplify matters by applying the triangle inequality to separate the translational and rotational components, which are in turn bounded in Lemmas 3.3.2 and 3.3.3.

$$\begin{aligned} A_{sym}(\rho) &\leq \max_{\alpha} \left( \max_{\vec{t}\in\mathcal{T}_\rho(\alpha)} \|\vec{t}\| + \max_{\vec{r}\in\mathcal{R}_\rho(\alpha)} \|\vec{r}\| \right) + 1 \\ &\leq \max_{\alpha} \left( \sqrt{1-\rho^2\sin^2\alpha} + 2\sqrt{3}\rho\sin\frac{\alpha}{2} \right) + 1. \end{aligned}$$

Substituting $x = \cos\alpha$, it follows that the quantity to be maximized is

$$f_\rho(x) \;=\; \sqrt{1-\rho^2(1-x^2)} + \sqrt{6}\rho\sqrt{1-x} + 1, \qquad \text{where } \sqrt{1-\frac{1}{\rho^2}} \leq x \leq 1.$$

To determine the maximum value of $f_\rho$ we take the partial derivative with respect to $x$, yielding

$$\frac{\partial f_\rho}{\partial x} \;=\; -\frac{\sqrt{3}\rho}{\sqrt{2(1-x)}} + \frac{\rho^2 x}{\sqrt{1-\rho^2(1-x^2)}}.$$

By our assumption that $\rho \geq \rho^*_{sym} \approx 1.26$, it follows by symbolic manipulations that $\partial f_\rho/\partial x = 0$ has a single real root for $\rho \geq \rho^*_{sym}$, which is

$$\begin{aligned} x_0(\rho) &= \frac{1}{6}\left(-1 + c_1(\rho) + \frac{1}{c_1(\rho)}\right), \\ &\quad \text{where} \quad c_1(\rho) = \frac{\rho^2}{(161\rho^6 + 18\rho^4\sqrt{c_2(\rho)} - 162\rho^4)^{1/3}} \\ &\quad \text{and} \quad c_2(\rho) = 80\rho^4 - 161\rho^2 + 81. \end{aligned}$$

By an analysis of this derivative it follows that the function $f_\rho$ achieves its maximum value when $x = x_0(\rho)$. Thus, we have the following.

**Lemma 3.3.4** *For $\rho \geq \rho^*_{sym} \approx 1.26$ we have*

$$A_{sym}(\rho) \;\leq\; f_\rho(x_0(\rho)) \;=\; \sqrt{1-\rho^2(1-x_0(\rho)^2)} + \sqrt{6}\rho\sqrt{1-x_0(\rho)} + 1,$$

*where $x_0(\rho)$ is defined above.*

Unfortunately, this function is too complex to reason about easily. Nonetheless, we can evaluate it numerically for any fixed value of $\rho$. The resulting approximation ratio (together with the alternate approximation ratio of Lemma 3.3.1) is plotted as a function of $\rho$ in Fig. 6 below. A numerical analysis shows that the two functions cross over at $\rho^*_{sym} \approx 1.26$, and at this point the function achieves its maximum value of roughly 3.14. The figure also indicates that the approximation ratio approaches 3 as $\rho$ tends to $\infty$. The following result establishes this asymptotic convergence by proving a somewhat weaker approximation ratio.

**Lemma 3.3.5** *If $\rho > \rho^*_{sym}$ then $A_{sym}(\rho) \leq 3 + \frac{1}{\sqrt{3}\rho}$.*

**Proof**: Before presenting the general case, we consider the simpler limiting case when $\rho$ tends to $\infty$. Since $0 \leq \sin\alpha \leq 1/\rho$, in the limit $\alpha$ approaches 0. Using the fact of $\lim_{\alpha\to 0} \frac{\sin\alpha}{\alpha} = 1$ we have

$$\begin{aligned} A_{sym}(\infty) \;&=\; \lim_{\rho\to\infty} A_{sym}(\rho) \;\leq\; \lim_{\rho\to\infty} \sqrt{1-\rho^2\sin^2\alpha} + 2\sqrt{3}\rho\sin\frac{\alpha}{2} + 1 \\ &=\; \lim_{\rho\to\infty} \sqrt{1-\rho^2\alpha^2} + \sqrt{3}\rho\alpha + 1. \end{aligned}$$

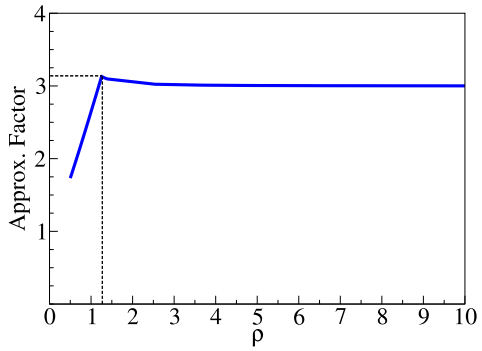


Fig. 6: The approximation ratio for the symmetric alignment algorithm as a function of $\rho$.

Let $x = \rho\alpha$. In the limit we have $0 \leq x \leq 1$ and so

$$A_{sym}(\infty) \;\leq\; \max_{0\leq x\leq 1}\left(\sqrt{1-x^2}+\sqrt{3}x+1\right).$$

It is easy to verify that $A_{sym}(\infty)$ achieves a maximum value of 3 when $x = \sqrt{3}/2$.

Next, we consider the general case. To simplify $A_{sym}(\rho)$, we apply a Taylor's series expansion.

$$\begin{aligned} A_{sym}(\rho) &\leq \max_{\alpha}\left(\sqrt{1-\rho^2\sin^2\alpha}+2\sqrt{3}\rho\sin\frac{\alpha}{2}+1\right) \\ &\leq \max_{\alpha}\left(\sqrt{1-\rho^2\left(\alpha-\frac{\alpha^3}{6}\right)^2}+\sqrt{3}\rho\alpha+1\right). \end{aligned} \tag{3.1}$$

Since $0 \leq \alpha \leq \arcsin(1/\rho)$, we have $0 \leq \rho\alpha \leq \rho\arcsin(1/\rho)$. For all $\rho > \rho^*_{sym}$, observe that $\rho\arcsin(1/\rho)$ is at most $\rho^*_{sym}\arcsin(1/\rho^*_{sym})$. Let us denote the value by $c_0$, which by numerical evaluation is approximately 1.16. Thus, we have $0 \leq \rho\alpha < c_0$. The rest of the analysis is broken into two cases: $0 \leq \rho\alpha \leq 1$ and $1 < \rho\alpha < c_0$.

In the first case ($0 \leq \rho\alpha \leq 1$) observe that $1 - \rho^2\alpha^2 \geq 0$. We argued above

that $(\sqrt{1-\rho^2\alpha^2}+\sqrt{3}\rho\alpha+1) \leq 3$ and so

$$\begin{aligned} A_{sym}(\rho) &\leq \max_{\alpha}\left(\sqrt{1-\rho^2\left(\alpha-\alpha^3/6\right)^2}+\sqrt{3}\rho\alpha+1+\left[3-\left(\sqrt{1-\rho^2\alpha^2}+\sqrt{3}\rho\alpha+1\right)\right]\right) \\ &= \max_{\alpha}\left(\sqrt{1-\rho^2\left(\alpha-\alpha^3/6\right)^2}-\sqrt{1-\rho^2\alpha^2}+3\right). \end{aligned}$$

To simplify this let $x=\rho\alpha$. We see that $A_{sym}(\rho) \leq \max_{0\leq x\leq 1} g(x)$ where,

$$g(x) \;=\; \sqrt{1-x^2\left(1-\frac{x^2}{6\rho^2}\right)^2}-\sqrt{1-x^2}+3.$$

For all fixed $\rho$ this is a monotonically increasing function in $x$. Since $x \leq 1$, it is easy to verify that this function achieves its maximum value at $x=1$. Therefore,

$$A_{sym}(\rho) \;\leq\; g(1) \;=\; 3+\sqrt{1-\left(1-\frac{1}{6\rho^2}\right)^2} \;=\; 3+\sqrt{\frac{1}{3\rho^2}-\frac{1}{36\rho^4}} \;\leq\; 3+\frac{1}{\sqrt{3}\rho},$$

as desired.

For the second case $(1<\rho\alpha<c_0)$ we cannot use the above approach because $1-\rho^2\alpha^2<0$. Nonetheless, by expanding the first term of Eq. (3.1) we have

$$\begin{aligned} \sqrt{1-\rho^2\sin^2\alpha} &\leq \sqrt{1-\rho^2\left(\alpha-\alpha^3/6\right)^2} \;=\; \sqrt{(1-\rho^2\alpha^2)+\frac{1}{3}\rho^2\alpha^4-\frac{1}{36}\rho^2\alpha^6} \\ &< \sqrt{\frac{1}{3}\rho^2\alpha^4} \;=\; \frac{1}{\sqrt{3}}\rho\alpha^2. \end{aligned}$$

Therefore,

$$A_{sym}(\rho) \;\leq\; \max_{\alpha}\left(\frac{1}{\sqrt{3}}\rho\alpha^2+\sqrt{3}\rho\alpha+1\right).$$

It is straightforward to verify that the above function achieves its maximum value when $\rho\alpha=c_0$, and this value is at most $3+1/(\sqrt{3}\rho)$. □

Combining this with Lemma 3.3.1 and the previous comments about the cross-over point completes the proof of Theorem 5.

## 3.4 Symmetric Alignment: Lower Bound

It is natural to wonder whether the upper bound on the approximation ratio $A_{sym}$ proved in Theorem 5 is tight. We show that this is nearly the case. In particular, we show that for all sufficiently large $\rho$ the approximation factor is strictly greater than 3, and it approaches 3 in the limit.

**Theorem 6** *For all sufficiently large $\rho$, there exists an input on which symmetric alignment achieves an approximation ratio of at least $3 + \frac{1}{10\rho^2}$.*

The remainder of this section is devoted to proving this. Our approach is to consider a configuration of points that generates the worst case (that is, the largest translation and rotational displacements) for Lemmas 3.3.2 and 3.3.3. Consider a fixed value of $\rho$. We define the pattern point set $P = \{p_1, p_2, p_3, p_4\}$ as follows. The first three points form an equilateral triangle of side length $2\rho$ oriented in counterclockwise order. We place $p_4$ at the midpoint of $\overline{p_1 p_2}$. (See Fig. 7(a)) By an infinitesimal perturbation of the points, we may assume that the pair $(p_1, p_2)$ is the unique diametrical pair for $P$.

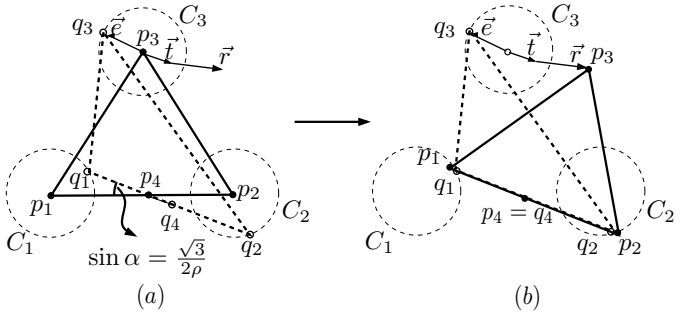


Fig. 7: The lower bound on $A_{sym}$.

We define the background set $Q = \{q_1, q_2, q_3, q_4\}$ so that the optimum Hausdorff distance will be at most 1. For $1 \leq i \leq 4$, let $C_i$ be a circle of unit radius centered $p_i$. Consider an angle $\alpha > 0$ such that $\sin\alpha = \frac{\sqrt{3}}{2\rho}$ and construct a line passing through $p_4$ (the midpoint of $\overline{p_1p_2}$) forming an angle $\alpha$ with line $\overline{p_1p_2}$. Place $q_1$ and $q_2$ at the rightmost intersection points of this line and $C_1$ and $C_2$, respectively. (See Fig. 7(a).) Running symmetric alignment so that $(p_1, p_2)$ is aligned with $(q_1, q_2)$ results in a translation and rotation. Let $\vec{t}$ and $\vec{r}$ denote the translation and rotational displacement vectors for $p_3$, respectively. Let $q_3 = p_3 + \vec{e}$, where $\vec{e}$ is a vector of unit length whose direction is chosen to be directly opposite that of $\vec{t} + \vec{r}$. Intuitively, $q_3$ has been chosen to be as far away as possible from $p_3$ after alignment. Finally, place $q_4$ at the midpoint of $\overline{q_1q_2}$. Observe that prior to alignment, each $p_i$ is within distance 1 from $q_i$, and therefore $h_{opt} \leq 1$.

Consider the placement of $P$ and $Q$ after symmetric alignment. (See Fig. 7(b).) We will analyze the displacement distance of $p_3$ relative to $q_3$. Because of the asymmetry introduced by $p_4$ and $q_4$, it is intuitively clear that if the distance ratio is high enough, symmetric alignment will achieve the best match by aligning the pair $(q_1, q_2)$ with $(p_1, p_2)$. We establish this formally in the next lemma.

**Lemma 3.4.1** *For all sufficiently large $\rho$, symmetric alignment aligns $(p_1, p_2)$ with $(q_1, q_2)$.*

**Proof**: Assume that $\rho > 2A_{sym}$. We will show that the choice of any other alignment pair will result in a Hausdorff distance greater than $A_{sym}h_{opt}$, a contradiction. Suppose that the algorithm aligns $(p_1, p_2)$ with some pair $(q_i, q_j)$, where

$(q_i, q_j) \neq (q_1, q_2)$. For $1 \leq i \leq 4$, let $C_i$ denote the circle of radius $A_{sym}$ centered at point $p_i$, after alignment. (These are different from the circles $C_i$ used in the construction.) Since $\rho > 2A_{sym}$, all these circles are disjoint. In order for the Hausdorff distance to be less than $A_{sym}$, after alignment each of these circles must contain exactly one point of $Q$.

First, we will show that $q_4$ should be in $C_4$. Suppose that to the contrary that $q_4$ is in circle $C_k$, $k \neq 4$. Then $q_1$ must lie in some other circle. Since $q_1$ and $q_2$ are symmetric with respect to the point $q_4$, the point $q_2$ cannot be in any circle, a contradiction.

From the hypothesis that $(p_1, p_2) \neq (q_1, q_2)$, either $q_1$ or $q_2$ should be in $C_3$. Let us assume that $q_1$ is in $C_3$. (The other case is similar.) Then $q_2$ may be in either $C_1$ or $C_2$. However, if $q_2$ lies in $C_2$, then clearly $q_3$ lies in none of the circles. (Note that we do not allow for reflection.) Thus, $q_2$ must lie within $C_1$ and $q_3$ lies within $C_2$, implying that $(q_2, q_3)$ is the chosen aligning pair for $(p_1, p_2)$. Thus, (after alignment) the midpoint of $\overline{q_2 q_3}$, called it $m_{23}$, coincides with $p_4$. Since $\triangle q_4 q_2 m_{23}$ and $\triangle q_1 q_2 q_3$ are similar up to a scale factor of 2, we have

$$\|p_4 q_4\| \;=\; \frac{\|q_1 q_3\|}{2}.$$

Also, observe that prior to alignment, each point of $Q$ is within distance $h_{opt}$ of its corresponding point of $P$, and so by the triangle inequality the distance between any two points $\|q_i q_j\|$ of $Q$ is within $2h_{opt}$ of the distance $\|p_i p_j\|$. So $\|q_1 q_3\| \geq \|p_1 p_3\| - 2h_{opt}$. Since the transformation is rigid, this is true after alignment. Since

$\|p_1p_3\| = 2\rho$, we have

$$\|p_4q_4\| \;\geq\; \frac{\|p_1p_3\| - 2h_{opt}}{2} \;=\; \frac{2\rho - 2h_{opt}}{2} \;=\; \rho - h_{opt}.$$

This exceeds $A_{sym}h_{opt}$ for all sufficiently large $\rho$, yielding the desired contradiction.

□

To complete the analysis of the lower bound, let us consider the translational and rotational displacement distances of $p_3$ that result from symmetric alignment. The line segment $\overline{q_1q_2}$ passes through $p_4$ (the midpoint of $\overline{p_1p_2}$). Let $\overline{q_2'q_2}$ denote the intersection of the line segment $\overline{q_1q_2}$ with the circle $C_2$. By simple trigonometry $\|q_2'q_2\| = 2\sqrt{1-\rho^2\sin^2\alpha}$, which equals 1 by our choice of $\alpha$. It is easy to see that the translation distance is half the chord length $\|q_2'q_2\|$ along a direction at angle $-\alpha$. Thus the translational displacement vector is

$$\vec{t} \;=\; \frac{1}{2}\|q_2q_2'\|\left(\cos\left(-\alpha\right), \sin\left(-\alpha\right)\right) \;=\; \frac{1}{2}\left(\cos\left(-\alpha\right), \sin\left(-\alpha\right)\right).$$

Next, let us consider the displacement of $p_3$ due to rotation. The rotation is about $p_4$, the midpoint of $p_1$ and $p_2$, and the angle of rotation is $\alpha$. Since $\triangle p_1p_2p_3$ is an equilateral triangle of side length $2\rho$, the distance from the center of rotation to $p_3$ is $\sqrt{3}\rho$. From simple trigonometry, the displacement distance due to rotation is $\sqrt{3}\rho \cdot 2\sin\frac{\alpha}{2}$ and the direction of the displacement is $-\frac{\alpha}{2}$. Thus the rotational displacement vector is

$$\vec{r} \;=\; 2\sqrt{3}\rho\sin\frac{\alpha}{2}\left(\cos\left(-\frac{\alpha}{2}\right), \sin\left(-\frac{\alpha}{2}\right)\right).$$

Observe that the angle between $\vec{t}$ and $\vec{r}$ is $\alpha/2$.

To complete the analysis, we decompose the rotational displacement vector $\vec{r}$ into two components, $\vec{r}_1$ is parallel to $\vec{t}$ and $\vec{r}_2$ is perpendicular to it. we have

$$\begin{aligned}\|\vec{r}_1\| &= 2\sqrt{3}\rho \sin\frac{\alpha}{2}\cos\frac{\alpha}{2} = \sqrt{3}\rho\sin\alpha = \frac{3}{2}\\ \|\vec{r}_2\| &= 2\sqrt{3}\rho\sin^2\frac{\alpha}{2} = \sqrt{3}\rho(1-\cos\alpha) = \sqrt{3}\rho\left(1-\sqrt{1-\frac{3}{4\rho^2}}\right).\end{aligned}$$

The squared length of the displacement vector is $D = \|\vec{t}+\vec{r}_1+\vec{r}_2\|^2 = \left(\|\vec{t}\| + \|\vec{r}_1\|\right)^2 + \|\vec{r}_2\|^2$. Thus, we have

$$D = \left(\frac{1}{2}+\frac{3}{2}\right)^2 + \|\vec{r}_2\|^2 = 4 + 3\rho^2\left(1-\sqrt{1-\frac{3}{4\rho^2}}\right)^2.$$

Using the fact that $\sqrt{1-x} \le 1-\frac{x}{2}$ we have

$$D \ge 4 + 3\rho^2\left(1-\left(1-\frac{3}{8\rho^2}\right)\right)^2 = 4+\frac{27}{64\rho^2}.$$

We may assume that $\rho \ge 2$, from which it follows that

$$\frac{27}{64\rho^2} \ge \frac{4}{10\rho^2}+\frac{1}{400\rho^2} \ge \frac{4}{10\rho^2}+\frac{1}{100\rho^4}.$$

Thus, we obtain

$$D \ge 4+\frac{4}{10\rho^2}+\frac{1}{100\rho^4} = \left(2+\frac{1}{10\rho^2}\right)^2 \qquad \text{and so} \qquad \sqrt{D} \ge 2+\frac{1}{10\rho^2}.$$

By adding the initial distance of 1 from $q_3$ to $p_3$ to the displacement, it follows that the final distance from $p_3$ to $q_3$ is $1+\sqrt{D}$. Combining this with the fact that $h_{opt} \le 1$, the approximation ratio satisfies

$$A_{sym}(\rho) \ge \frac{1+\sqrt{D}}{h_{opt}} \ge 3+\frac{1}{10\rho^2}.$$

This completes the proof of Theorem 6. Clearly this is greater than 3 for all positive $\rho$ and converges to 3 in the limit as $\rho$ tends to $\infty$.

## 3.5 Serial Alignment: Upper Bound

In this section we derive an upper bound on the approximation ratio $A_{ser}(\rho)$ for serial alignment as a function of the distance ratio $\rho$. (Recall the definitions presented at the start of Section 3.3.) The main result of this section is presented below.

**Theorem 7** *Consider two planar point sets $P$ and $Q$, and let $\rho$ be the distance ratio for $P$ and $Q$. Then*

$$A_{ser}(\rho) \;\leq\; \min\left(c_\infty + \frac{9}{4\rho},\, 2\rho\right),$$

*where $c_\infty$ is defined to be $\lim_{\rho\to\infty} A_{ser}(\rho) \approx 3.196$. Further $A_{ser}(\rho) \leq 3.44$ for all $\rho$.*

As in the analysis of symmetric alignment we will establish two approximation ratios as functions of $\rho$, one is better for low $\rho$ and the other for high $\rho$. We will show that the two functions cross over at a value $\rho^*_{ser} \approx 1.72$ and that $A_{ser}(\rho^*_{ser}) \approx 3.44$. Furthermore, both these functions decrease monotonically as $\rho$ moves away from $\rho^*_{ser}$. The approximation ratio for low distance ratios is presented in Lemma 3.5.1 below. As usual, assume that $P$ has been transformed to its optimal placement with respect to $Q$. Let $(p_1, p_2)$ denote the diametrical pair in $P$ and let $(q_1, q_2)$ denote the corresponding pair $Q$. Let $\alpha \geq 0$ denote the absolute acute angle between the lines $\overline{p_1p_2}$ and $\overline{q_1q_2}$.

**Lemma 3.5.1** *For all $\rho$, $A_{ser}(\rho) \leq 2\rho$.*

**Proof**: After applying serial alignment $p_1$ and $q_1$ coincide. Thus, after alignment the distance between any point $p \in P$ and $q_1$ is at most $2\rho$ since the distance from

$p_1$ to $p$ is at most $2\rho$. Therefore, the Hausdorff distance after alignment is at most $2\rho$. □

To deal with the more interesting case of high $\rho$ we consider the effects of translation and rotation separately as functions of $\rho$ and the rotation angle $\alpha$. These results are presented Sections 3.5.1 and 3.5.2, respectively. Unlike the corresponding analysis for symmetric alignment of Section 3.3, the analysis here will require a more detailed understanding of the geometric structure of the translation and rotation spaces.

### 3.5.1 Translational Displacement

Let us consider the space of possible translation vectors that align $p_1$ with $q_1$. Since $P$ is in its optimal placement, it follows that $q_1$ and $q_2$ lie within two circles $C_1$ and $C_2$ of radius $h_{opt}$ $(= 1)$ centered at $p_1$ and $p_2$, respectively. (See Fig. 8(a).) For the sake of illustration, let us assume that $\overrightarrow{p_1p_2}$ is directed horizontally from left to right as shown in the figure and that the line $\overline{q_1q_2}$ has a nonpositive slope (that is, $\alpha$ is a clockwise angle). Consider a line $\ell$ that is parallel to $\overline{q_1q_2}$ and is tangent to $C_2$ from below. This line forms an angle of $\alpha$ with respect to $\overline{p_1p_2}$ and clearly it intersects $C_1$.

Let $h_1$ denote the orthogonal projection of $p_1$ onto the line $\ell$, and let $h_2$ denote the orthogonal projection of $p_2$ onto the line passing through $p_1$ and parallel to $\ell$. By simple trigonometry $\|p_2h_2\| = \|p_1p_2\| \sin\alpha$ and since $C_2$'s radius is 1, the signed distance from $p_1$ to $h_1$ is $2\rho \sin\alpha - 1$. The interior of circle $C_1$ lying above $\ell$ defines

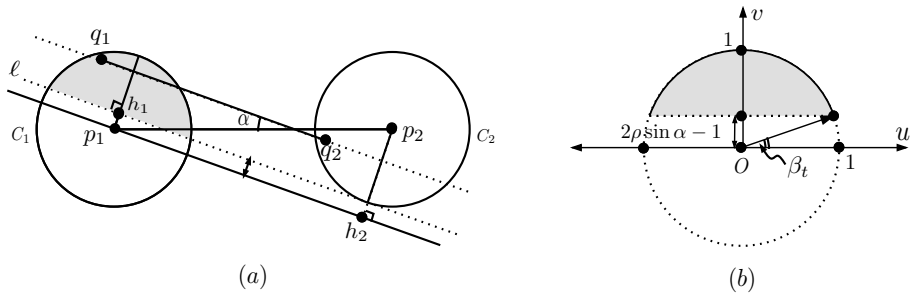


Fig. 8: The translation space for serial alignment.

a region called a *segment* of $C_1$. (This is the shaded region of Fig. 8(a).) Clearly, $q_1$ lies within this segment. It follows that the set of possible translation vectors $\overrightarrow{p_1q_1}$ is a copy of this region translated so that $p_1$ coincides with the origin.

As in Section 3.3.1, we introduce $(u, v)$ coordinate system by rotating original coordinate system by $\alpha$. In particular, the $u$-axis is parallel to $\overline{q_1q_2}$, and the $v$-axis is orthogonal and directed upwards. This yields the following characterization of the translation space in terms of this coordinate system, which is illustrated in Fig. 8(b). In this section and the next it will simplify notation somewhat to use polar coordinates to represent vectors. The polar coordinates $\langle s, \phi \rangle$ indicate that $s$ is the length of the vector, and $\phi$ is a counterclockwise angle with respect to the positive horizontal axis.

**Lemma 3.5.2** *For fixed $\alpha$ and $\rho$, the translation space $\mathcal{T}_\rho(\alpha)$ in $(u, v)$ coordinates is the segment of a circle of radius $h_{opt}$ $(= 1)$ centered the origin that lies above the line $v = 2\rho\sin\alpha - 1$. The rightmost vertex of this segment has polar coordinates $\langle 1, \beta_t \rangle$, where $\sin\beta_t = 2\rho\sin\alpha - 1$.*

For now we assume that $\beta_t > 0$. Later in Lemma 3.5.6 we will show that

this assumption is safe, since if $\beta_t \leq 0$ the resulting bound will be weaker than for positive $\beta_t$.

### 3.5.2 Rotational Displacement

Next we consider the effects of rotation on the approximation error. As in Section 3.3.2 we need to consider the placement of points in $P$ because the distance by which a point is displaced by rotation depends on the distance from this point to the center of rotation. For given $\rho$ and $\alpha$ values, let $\mathcal{R}_\rho(\alpha)$ denote the set of possible rotational displacement vectors.

Because $(p_1, p_2)$ is a diametrical pair, every point of $P$ lies within a lune formed by the intersection of two discs of radius $2\rho$ centered at $p_1$ and $p_2$, respectively. (See Fig. 9(a).) As before, for the sake of illustration let us assume that $\overrightarrow{p_1p_2}$ is directed horizontally from left to right as shown in the figure, and that $\alpha$ is a clockwise angle. Along any fixed direction, the farther that a point is from the center of rotation $p_1$ the greater its rotational displacement is. So we will concentrate on the most distant points from the center of rotation, that is, points lying on the boundary of this lune. For any angle $\theta$, where $-\pi/2 \leq \theta \leq \pi/2$, let $p_\theta$ denote the point on this lune such that the signed angle $\angle p_2p_1p_\theta$ is $\theta$. (See Fig 9(a).) Let $p'_\theta$ denote this point's position after a clockwise rotation about $p_1$ by angle $\alpha$. Let $\vec{r}(\theta) = \overrightarrow{p_\theta p'_\theta}$ be the corresponding rotational displacement vector.

The following lemma characterizes $\mathcal{R}_\rho(\alpha)$ by describing $\vec{r}(\theta)$ as a function of $\theta$. This region is illustrated in Fig. 9(b).

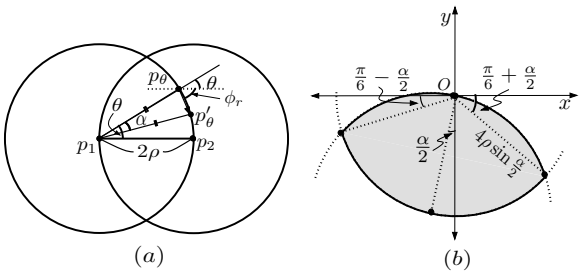


Fig. 9: The rotation space for serial alignment.

**Lemma 3.5.3** *Consider fixed $\alpha$ and $\rho$, and let*

$$s_0 \;=\; 4\rho \sin \frac{\alpha}{2}, \qquad \phi_1 \;=\; -\left(\frac{5\pi}{6} + \frac{\alpha}{2}\right), \qquad \textit{and} \qquad \phi_2 \;=\; -\left(\frac{\pi}{6} + \frac{\alpha}{2}\right).$$

*The rotational displacement space $\mathcal{R}_\rho(\alpha)$ is a region bounded by two circular arcs:*

(i) *the arc of a circle centered at the origin of radius $s_0$ for $\phi \in [\phi_1, \phi_2]$.*

(ii) *the arc of a circle passing through the origin whose radius is $s_0$, and whose center lies at the polar coordinates $\left\langle s_0, -\frac{\pi+\alpha}{2} \right\rangle$, for $\phi \in [\phi_2, 0] \cup [-\pi, \phi_1]$.*

*Furthermore, for all $\vec{r} \in R_\rho(\alpha)$, we have $\|\vec{r}\| \leq s_0$.*

**Proof**: Throughout we will use the following simple observation. The triangle $\triangle p_1 p_\theta p'_\theta$ is isosceles and its apex angle is $\alpha$. Thus, the length of the displacement vector $\overrightarrow{p_\theta p'_\theta}$ is the length of the triangle's base, which is $2\|p_1 p_\theta\| \sin \frac{\alpha}{2}$, and its angle with respect to $\overrightarrow{p_1 p_2}$ is $-\left(\frac{\pi}{2} - \theta + \frac{\alpha}{2}\right)$. We consider two cases depending on $\theta$, where $-\pi/2 \leq \theta \leq \pi/2$.

(i) If $\theta \in [-\pi/3, \pi/3]$ then $p_\theta$ lies on the opposite side of the lune from $p_1$, that is, on the arc of the circle of radius $2\rho$ centered at $p_1$. Because all such points are at distance $2\rho$ from $p_1$, by the above observation, the length of the

displacement vector for all such points is $4\rho \sin \frac{\alpha}{2} = s_0$, and the direction is $\phi = -\left(\frac{\pi}{2} - \theta + \frac{\alpha}{2}\right)$. For the given range of $\theta$ we have

$$\phi \in \left[-\left(\frac{\pi}{2} + \frac{\pi}{3} + \frac{\alpha}{2}\right), -\left(\frac{\pi}{2} - \frac{\pi}{3} + \frac{\alpha}{2}\right)\right] = \left[-\left(\frac{5\pi}{6} + \frac{\alpha}{2}\right), -\left(\frac{\pi}{6} + \frac{\alpha}{2}\right)\right] = [\phi_1, \phi_2].$$

Clearly this is a circular arc centered at the origin of radius $s_0$.

(ii) If $\theta \in [\pi/3, \pi/2] \cup [-\pi/2, -\pi/3]$ then $p_\theta$ lies on the left boundary of the lune passing through $p_1$. This is the arc of the circle of radius $2\rho$ centered at $p_2$. By simple trigonometry we have

$$\|p_1 p_\theta\| = 2\rho \cdot 2 \sin \frac{\pi - 2\theta}{2} = 4\rho \cos \theta.$$

By the above observation, the length of the rotational displacement vector is $8\rho \cos\theta \sin \frac{\alpha}{2}$ and its direction is $\phi = -\left(\frac{\pi}{2} - \theta + \frac{\alpha}{2}\right)$. Therefore, as a function of $\theta$, the polar coordinates of the displacement vector are

$$\left\langle 8\rho \cos\theta \sin \frac{\alpha}{2}, -\left(\frac{\pi}{2} - \theta + \frac{\alpha}{2}\right)\right\rangle.$$

It is easy to verify algebraically the resulting curve is the arc of a circle passing through the origin whose radius is $4\rho \sin \frac{\alpha}{2} = s_0$, and whose center lies at the polar coordinates

$$\left\langle 4\rho \sin \frac{\alpha}{2}, -\frac{\pi + \alpha}{2}\right\rangle = \left\langle s_0, -\frac{\pi + \alpha}{2}\right\rangle.$$

□

### 3.5.3 Combining Translation and Rotation

Finally we use the results of the prior two sections to derive the approximation ratio for the serial alignment algorithm. The space of possible translational displacement

vectors, $\mathcal{T}_\rho(\alpha)$, was shown in Fig. 8 and space of possible rotational displacement vectors, $\mathcal{R}_\rho(\alpha)$, was shown in Fig. 9. The former was shown in $(u, v)$ coordinates (aligned with $\overline{q_1q_2}$) and the latter was shown in $(x, y)$ coordinates (aligned with $\overline{p_1p_2}$). We show them both in Fig. 10 in $(u, v)$ coordinates by rotating the latter counterclockwise by angle $\alpha$.

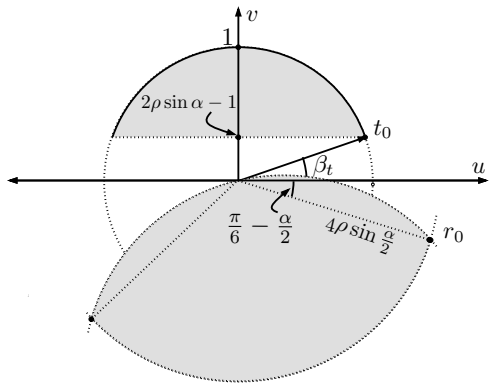


Fig. 10: Translation and rotation space for serial alignment.

To derive the overall approximation bound it suffices to find (as a function of $\rho$ and maximizing over all $\alpha$) the two vectors $\vec{t} \in \mathcal{T}_\rho(\alpha)$ and $\vec{r} \in \mathcal{R}_\rho(\alpha)$ whose sum produces the largest total displacement distance $\|\vec{t} + \vec{r}\|$. Because each point of $P$ is initially within distance $h_{opt}$ $(= 1)$ of its closest point of $Q$, and so the final approximation bound will be $\|\vec{t} + \vec{r}\| + 1$. In order to do this, we need to determine the pair of vectors from these spaces whose sum achieves the greatest distance. A visual inspection of Fig. 10 suggests that the desired pair consists of the rightmost vertex of each region. We establish this in the following lemma.

Recall from Lemma 3.5.2 the angle $\beta_t$ where $\sin \beta_t = 2\rho \sin \alpha - 1$, which the angle to the rightmost vertex of $\mathcal{T}_\rho(\alpha)$. Recall from the comments after Lemma 3.5.2 that we may assume that $\beta_t > 0$. Let $\vec{t}_0$ denote the unit length vector to this vertex,

whose polar coordinates are $\langle 1, \beta_t \rangle$. Also recall from Lemma 3.5.3 that the angle (in $(x, y)$ coordinates) to the rightmost vertex of $\mathcal{R}_\rho(\alpha)$ is $\phi_2 = -\left(\frac{\pi}{6} + \frac{\alpha}{2}\right)$. To convert this into $(u, v)$ coordinates we add $\alpha$, and so let us define $\phi_r = \phi_2 + \alpha = -\frac{\pi}{6} + \frac{\alpha}{2}$. Let $\vec{r}_0 \in \mathcal{R}_\rho(\alpha)$ denote the vector to this vertex whose polar coordinates are $\left\langle 4\rho \sin\frac{\alpha}{2}, \phi_r \right\rangle$.

**Lemma 3.5.4** *Assuming $\beta_t > 0$,*

$$\max_{\substack{\vec{t} \in \mathcal{T}_\rho(\alpha) \\ \vec{r} \in \mathcal{R}_\rho(\alpha)}} \|\vec{t} + \vec{r}\|^2 \;\leq\; 1 + \left(4\rho \sin\frac{\alpha}{2}\right)^2 + 8\rho \sin\frac{\alpha}{2} \sin\left(\frac{\pi}{3} - \beta_t + \frac{\alpha}{2}\right).$$

*The sum of the vectors $\vec{t}_0$ and $\vec{r}_0$ achieves this maximum length.*

**Proof**: We will show that for all $\vec{t} \in \mathcal{T}_\rho(\alpha)$ and $\vec{r} \in \mathcal{R}_\rho(\alpha)$, $\|\vec{t} + \vec{r}\| \leq \|\vec{t}_0 + \vec{r}_0\|$. First we observe that $\phi_r < 0$. To see this recall that $\rho > \rho^*_{ser}$ and $\sin\alpha < 1/\rho$. Thus

$$\sin\alpha \;<\; \frac{1}{\rho^*_{ser}} \;\approx\; \frac{1}{1.72} \;<\; \frac{\sqrt{3}}{2} \;=\; \sin\frac{\pi}{3}.$$

Therefore $\alpha < \pi/3$, implying that $\phi_r < 0$.

Assuming that the angle between the two vectors $\vec{t}$ and $\vec{r}$ is acute, in order to maximize $\|\vec{t} + \vec{r}\|$ we should maximize the lengths of these vectors while minimizing the angle between them. Because the upper boundary of $\mathcal{T}_\rho(\alpha)$ is a circle centered at the origin, it is easy to see that the optimal choice would be at either the leftmost or rightmost vertices of the region. Because of the asymmetry introduced by $\alpha$, the rightmost vertex $t_0$ is easily seen to be the better choice. (This is based on our assumption that $\alpha$ is a clockwise angle. The leftmost vertex would be chosen if the rotation had been counterclockwise.)

The remaining problem is to determine the best choice of $\vec{r}$. The lower boundary of $\mathcal{R}_\rho(\alpha)$ is a circular arc centered at the origin, and so the optimal choice cannot be in the interior of this arc. Thus, the optimal choice must lie along the upper boundary of $\mathcal{R}_\rho(\alpha)$, and in particular, along the arc running from the origin to $\vec{r}_0$. We will show that the optimum is achieved at $\vec{r}_0$. Recall from case (ii) of the proof of Lemma 3.5.3 that the points of this arc have polar coordinates (relative to $(x, y)$ coordinates)

$$\left\langle 8\rho\cos\theta\sin\frac{\alpha}{2}, -\left(\frac{\pi}{2} - \theta + \frac{\alpha}{2}\right)\right\rangle,$$

where $\pi/3 \le \theta \le \pi/2$. We can convert this to polar coordinates relative to $(u, v)$ coordinates by adding $\alpha$ to the angle. Let

$$\vec{r}(\theta) \;=\; \langle s(\theta), \phi(\theta)\rangle, \qquad \text{where} \quad s(\theta) = 8\rho\cos\theta\sin\tfrac{\alpha}{2}, \qquad \text{and} \quad \phi(\theta) = -\tfrac{\pi}{2} + \theta + \tfrac{\alpha}{2}.$$

Observe that $\vec{r}_0 = \vec{r}(\pi/3)$. Recall from Lemma 3.5.2 that the polar representation of $\vec{t}_0$ is $\langle 1, \beta_t\rangle$. By the law of cosines, the squared length of their sum can be expressed as a function $f(\theta) = \|\vec{t}_0 + \vec{r}(\theta)\|^2$, where

$$\begin{aligned} f(\theta) &= 1^2 + s(\theta)^2 - 2\cdot 1\cdot s(\theta)\cdot\cos\left(\pi - (\beta_t - \phi(\theta))\right) \\ &= 1 + \left(8\rho\cos\theta\sin\frac{\alpha}{2}\right)^2 + 16\rho\cos\theta\sin\frac{\alpha}{2}\sin\left(\theta - \beta_t + \frac{\alpha}{2}\right). \end{aligned}$$

To determine its maximum value, we consider the partial derivative of $f$ with respect to $\theta$.

$$\begin{aligned} \frac{\partial f}{\partial\theta} &= -128\rho^2\sin^2\frac{\alpha}{2}\cos\theta\sin\theta + \\ &\qquad 16\rho\sin\frac{\alpha}{2}\left(-\sin\theta\sin\left(\theta - \beta_t + \frac{\alpha}{2}\right) + \cos\theta\cos\left(\theta - \beta_t + \frac{\alpha}{2}\right)\right) \\ &= -128\rho^2\sin^2\frac{\alpha}{2}\cos\theta\sin\theta + 16\rho\sin\frac{\alpha}{2}\cos\left(2\theta - \beta_t + \frac{\alpha}{2}\right). \end{aligned}$$

To simplify notation let $w = 8\rho \sin \frac{\alpha}{2}$, and by simple trigonometry we have

$$\begin{aligned}
\frac{\partial f}{\partial \theta} &= -2w^2 \cos\theta \sin\theta + 2w \cos\left(2\theta - \beta_t + \frac{\alpha}{2}\right) \\
&= -w^2 \sin 2\theta + 2w \cos 2\theta \cos\left(\beta_t - \frac{\alpha}{2}\right) + 2w \sin 2\theta \sin\left(\beta_t - \frac{\alpha}{2}\right).
\end{aligned}$$

Now we will show that the derivative is negative under our assumption that $\pi/3 \leq \theta \leq \pi/2$ and $\beta_t > 0$. Observe that $\sin 2\theta \geq 0$ and $\cos 2\theta < 0$. To estimate the other terms we first consider the range of $\beta_t - \frac{\alpha}{2}$. Using the fact that $\sin \beta_t = 2\rho \sin\alpha - 1$ and $\beta_t > 0$ it follows that $0 < \beta_t \leq \pi/2$. Since $0 \leq \alpha < \pi/3$ for $\rho > \rho^*_{ser}$ we have

$$-\frac{\pi}{6} \;\leq\; \beta_t - \frac{\alpha}{2} \;\leq\; \frac{\pi}{2}.$$

Thus, $\cos(\beta_t - \alpha/2) \geq 0$. The only remaining term is $\sin(\beta_t - \alpha/2)$. If $\beta_t - \frac{\alpha}{2} \leq 0$ it is obvious that the partial derivative is negative. Otherwise, $\beta_t - \frac{\alpha}{2} > 0$, and so

$$\begin{aligned}
\frac{\partial f}{\partial \theta} &= -w^2 \sin 2\theta + 2w \cos 2\theta \cos\left(\beta_t - \frac{\alpha}{2}\right) + 2w \sin 2\theta \sin\left(\beta_t - \frac{\alpha}{2}\right) \\
&< -w^2 \sin 2\theta + 2w \sin 2\theta \;=\; (2-w) w \sin 2\theta.
\end{aligned}$$

Since $\beta_t > 0$ it follows that $2\rho \sin\alpha > 1$ and hence

$$w \;=\; 8\rho \sin\frac{\alpha}{2} \;>\; 4\rho \sin\alpha \;>\; 2.$$

Therefore, the partial derivative is negative for $\pi/3 \leq \theta \leq \pi/2$ and $\beta_t > 0$, and so $f(\theta)$ is a decreasing function of $\theta$. This implies that the maximum rotational displacement is achieved at the minimum value of $\theta = \frac{\pi}{3}$. Thus, the maximum combined displacement is achieved by the sum of $\vec{t}_0$ and $\vec{r}(\pi/3) = \vec{r}_0$, as desired. By plugging $\theta = \pi/3$ into $f(\theta)$ we obtain the maximum squared displacement distance value given in the statement of the lemma. □

Combining this result with our earlier remarks we have

$$\begin{aligned} A_{ser}(\rho) &\leq \max_{\alpha} \left( \max_{\substack{\vec{t}\in\mathcal{T}_\rho(\alpha) \\ \vec{r}\in\mathcal{R}_\rho(\alpha)}} \|\vec{t}+\vec{r}\| \right) + 1 \\ &= \max_{\alpha} \sqrt{1 + \left(4\rho \sin\frac{\alpha}{2}\right)^2 + 8\rho \sin\frac{\alpha}{2} \sin\left(\frac{\pi}{3} - \beta_t + \frac{\alpha}{2}\right)} + 1. \end{aligned}$$

Unfortunately, this function is too complex to reason about easily. However, the resulting approximation ratio can be computed numerically and plotted as a function of $\rho$. (See Fig 11.) By numerical means we determine that the cross-over point, denoted $\rho^*_{ser}$ is approximately 1.72, and $A_{ser}(\rho^*_{ser}) \leq 3.44$. The figure shows that as $\rho$ tends to $\infty$ the approximation bound decreases and converges to a value, which we denote by $c_\infty$. The next lemma provides a characterization of this function.

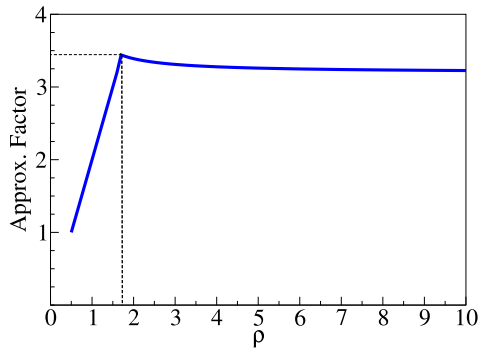


Fig. 11: The approximation ratio for serial alignment as a function of $\rho$.

**Lemma 3.5.5** *If* $\rho > \rho^*_{ser}$ *then* $A_{ser}(\rho) \leq c_\infty + \frac{9}{4\rho}$, *where* $c_\infty = \lim_{\rho\to\infty} A_{ser}(\rho) \approx 3.196$.

**Proof**: We will follow the approach we used in Lemma 3.3.5. We first find an optimal solution for the limiting case when $\rho \to \infty$ and then present the general

case. Let $D_\rho(\alpha)$ denote the maximum squared total displacement length as given in Lemma 3.5.4. When $\rho$ and $\alpha$ are clear from context we simply write $D$.

$$D \;=\; D_\rho(\alpha) \;=\; 1 + \left(4\rho \sin\frac{\alpha}{2}\right)^2 + 8\rho \sin\frac{\alpha}{2} \sin\left(\frac{\pi}{3} - \beta_t + \frac{\alpha}{2}\right).$$

As mentioned above, $A_{ser}(\rho) \le \max_\alpha \sqrt{D} + 1$. Since $0 \le \sin\alpha \le 1/\rho$, in the limit $\alpha$ approaches 0. Using the fact of $\lim_{\alpha\to 0} \frac{\sin\alpha}{\alpha} = 1$ and elementary trigonometry we have

$$\begin{aligned}
\lim_{\rho\to\infty} D &= \lim_{\rho\to\infty} \left(1 + \left(4\rho\sin\frac{\alpha}{2}\right)^2 + 8\rho\sin\frac{\alpha}{2}\left(\sin\left(\frac{\pi}{3}+\frac{\alpha}{2}\right)\cos\beta_t - \sin\beta_t\cos\left(\frac{\pi}{3}+\frac{\alpha}{2}\right)\right)\right) \\
&= \lim_{\rho\to\infty} 1 + (2\rho\alpha)^2 + 4\rho\alpha\left(\frac{\sqrt{3}}{2}\sqrt{4\rho\alpha - 4\rho^2\alpha^2} - (2\rho\alpha - 1)\frac{1}{2}\right) \\
&= \lim_{\rho\to\infty} 1 + 2\rho\alpha + 4\sqrt{3}\rho\alpha\sqrt{\rho\alpha - \rho^2\alpha^2}.
\end{aligned}$$

Substituting $x = \rho\alpha$ $(0 \le x \le 1)$, let

$$f(x) = 1 + 2x + 4\sqrt{3}x\sqrt{x - x^2}.$$

To find the value $x = x_0$ at which the maximum is achieved, we set the derivative to 0, from which we obtain

$$x_0 \;=\; \frac{1}{6}\left(3 + \sqrt{2}\cos\phi_0 + \sqrt{6}\sin\phi_0\right), \qquad \text{where } \; \phi_0 = \frac{1}{3}\arctan\frac{\sqrt{47}}{9}. \tag{3.2}$$

Observe that $x_0$ is slightly greater than $\sqrt{2/3}$. Thus, we have

$$c_\infty \;=\; A_{ser}(\infty) \;=\; \lim_{\rho\to\infty}\max_\alpha \sqrt{D_\rho(\alpha)} + 1 \;=\; \sqrt{f(x_0)} + 1 \;\approx\; 3.196. \tag{3.3}$$

Next, we consider the general case. Before considering $D$, let us simplify the last term of $D$ first. Let

$$g_\rho(\alpha) \;=\; \sin\left(\frac{\pi}{3} - \beta_t + \frac{\alpha}{2}\right).$$

From basic trigonometry we have

$$\begin{aligned} g_\rho(\alpha) &= \sin\left(\frac{\pi}{3}+\frac{\alpha}{2}\right)\cos\beta_t - \sin\beta_t\cos\left(\frac{\pi}{3}+\frac{\alpha}{2}\right) \\ &= \sin\frac{\pi}{3}\cos\frac{\alpha}{2}\cos\beta_t + \cos\frac{\pi}{3}\sin\frac{\alpha}{2}\cos\beta_t - \cos\frac{\pi}{3}\cos\frac{\alpha}{2}\sin\beta_t + \sin\frac{\pi}{3}\sin\frac{\alpha}{2}\sin\beta_t. \end{aligned}$$

To simplify $g_\rho(\alpha)$ we apply a Taylor's series expansion.

$$\begin{aligned} g_\rho(\alpha) &\leq \frac{\sqrt{3}}{2}\cos\beta_t + \frac{\alpha}{4}\cos\beta_t - \frac{1}{2}\left(1-\frac{\alpha^2}{8}\right)\sin\beta_t + \frac{\sqrt{3}\alpha}{4}\sin\beta_t \\ &\leq \left(\sqrt{3}+\frac{\alpha}{2}\right)\sqrt{\rho\alpha - \rho^2\left(\alpha-\frac{\alpha^3}{6}\right)^2} - \\ &\qquad \frac{1}{2}\left(1-\frac{\alpha^2}{8}\right)\left(2\rho\left(\alpha-\frac{\alpha^3}{6}\right)-1\right) + \frac{\sqrt{3}\alpha}{4}(2\rho\alpha-1). \end{aligned}$$

Now, applying a Taylor's series expansion of $D$ (as a function of $\alpha$) we have

$$\begin{aligned} D &= 1 + \left(4\rho\sin\frac{\alpha}{2}\right)^2 + 8\rho\sin\frac{\alpha}{2}\cdot g_\rho(\alpha) \\ &\leq 1 + (2\rho\alpha)^2 + 4\rho\alpha\cdot g_\rho(\alpha). \end{aligned}$$

Substituting $g_\rho(\alpha)$ for the last term of $D$ and removing all the negative terms we have

$$D \leq 1 + 2\rho\alpha + 4\rho\alpha\left(\left(\sqrt{3}+\frac{\alpha}{2}\right)\sqrt{\rho\alpha - \rho^2\alpha^2 + \frac{\rho^2\alpha^4}{3}} + \frac{7\rho\alpha^3}{24} + \frac{\sqrt{3}\rho\alpha^2}{2}\right) \quad (3.4)$$

Since $0 \leq \alpha \leq \arcsin(1/\rho)$, we have $0 \leq \rho\alpha \leq \rho\arcsin(1/\rho)$. For all $\rho > \rho^*_{ser}$, observe that $\rho\arcsin(1/\rho)$ is at most $\rho^*_{ser}\arcsin(1/\rho^*_{ser})$. Let us denote this value by $c_0$, which by numerical evaluation is approximately 1.07. Thus, we have $0 \leq \rho\alpha < c_0$.

We will show that $D \leq (c_\infty - 1)^2 + \frac{9}{\rho}$. We will consider two cases: $0 \leq \rho\alpha \leq 1$ and $1 < \rho\alpha < c_0$. For the first case ($0 \leq \rho\alpha \leq 1$) we substitute $x = \rho\alpha$ ($0 \leq x \leq 1$) and use the fact that $(1 + 2x + 4\sqrt{3}x\sqrt{x-x^2}) \leq f(x_0)$, which we argued above.

And so we have

$$\begin{aligned} D &\leq D + \left[ f(x_0) - \left( 1 + 2x + 4\sqrt{3}x\sqrt{x - x^2} \right) \right] \\ &= f(x_0) + 4\sqrt{3}x \left( \sqrt{x - x^2 + \frac{x^4}{3\rho^2}} - \sqrt{x - x^2} \right) + \\ &\qquad 4x \left( \frac{x}{2\rho} \sqrt{x - x^2 + \frac{x^4}{3\rho^2}} + \frac{7x^3}{24\rho^2} + \frac{\sqrt{3}x^2}{2\rho} \right). \end{aligned}$$

It is easily proved that this function is increasing for $0 \leq x \leq 1$. Thus, it achieves its maximum value at $x = 1$. By hypothesis, $\rho > \rho^*_{ser} > 1.72$, from which it follows that

$$D \leq f(x_0) + \left( \frac{4 + 2\sqrt{3}}{\rho} + \frac{7 + 4\sqrt{3}}{6\rho^2} \right) < f(x_0) + \frac{9}{\rho} = (c_\infty - 1)^2 + \frac{9}{\rho},$$

as desired.

For the second case ($1 < \rho\alpha < c_0$), since $\rho\alpha - \rho^2\alpha^2 < 0$, from Eq. (3.4) we have

$$D < 1 + 2\rho\alpha + 4\rho\alpha \left( \left( \sqrt{3} + \frac{\alpha}{2} \right) \sqrt{\frac{\rho^2\alpha^4}{3}} + \frac{7\rho\alpha^3}{24} + \frac{\sqrt{3}\rho\alpha^2}{2} \right).$$

It is easily observed that the function is an increasing function with respect to $\rho\alpha$ and the maximum value is achieved at $\rho\alpha = 1.07$, and this value is at most $(c_\infty - 1)^2 + \frac{9}{\rho}$, as desired.

Now, using the fact of $c_\infty > 3$ we have

$$D < (c_\infty - 1)^2 + \frac{9}{\rho} < \left( c_\infty - 1 + \frac{9}{4\rho} \right)^2.$$

Therefore,

$$A_{ser}(\rho) \leq \max_\alpha \sqrt{D} + 1 \leq c_\infty + \frac{9}{4\rho},$$

which completes the proof. □

The analysis of Lemma 3.5.4 was performed under the assumption that $\beta_t > 0$. We can justify this assumption now by showing that any other value would lead to a weaker approximation bound.

**Lemma 3.5.6** *For $\rho > \rho^*_{ser}$, if $\beta_t \leq 0$ then*

$$d_{ser}(\rho) \;<\; A_{ser}(\rho) h_{opt},$$

*where $d_{ser}(\rho)$ is a Hausdorff distance after running serial alignment.*

**Proof**: To simplify matters, assume that $h_{opt} = 1$ and let $d_{ser}(\rho) \leq \max_\alpha g_\rho(\alpha)$. By the triangle inequality we have

$$g_\rho(\alpha) \;=\; \max_{\substack{\vec{t}\in\mathcal{T}_\rho(\alpha)\\ \vec{r}\in\mathcal{R}_\rho(\alpha)}} (\|\vec{t}+\vec{r}\|+1) \;\leq\; \max_{\substack{\vec{t}\in\mathcal{T}_\rho(\alpha)\\ \vec{r}\in\mathcal{R}_\rho(\alpha)}} (\|\vec{t}\|+\|\vec{r}\|+1).$$

By Lemmas 3.5.2 and 3.5.3 we have $\|\vec{t}\| \leq 1$ and $\|\vec{r}\| \leq 4\rho \sin\frac{\alpha}{2}$, and so

$$g_\rho(\alpha) \;\leq\; 2 + 4\rho \sin\frac{\alpha}{2} \;=\; 2 + \frac{2\rho\sin\alpha}{\cos\frac{\alpha}{2}}.$$

By the assumption that $\beta_t \leq 0$, it follows $2\rho\sin\alpha - 1 \leq 0$, and so $g_\rho(\alpha) \;\leq\; 2 + \frac{1}{\cos(\alpha/2)}$. For $\rho > \rho^*_{ser}$ and $\sin\alpha \leq 1/(2\rho)$, we have

$$\cos\frac{\alpha}{2} \;=\; \frac{1}{\sqrt{2}}\sqrt{1+\sqrt{1-\sin^2\alpha}} \;>\; \frac{1}{\sqrt{2}}\sqrt{1+\sqrt{1-1/(2\rho^*_{ser})^2}} \;\approx\; \frac{1}{1.01}.$$

Thus, we see that the bound on $d_{ser}(\rho)$ generated under the hypothesis $\beta_t \leq 0$ is significantly smaller than $c_\infty$. In the proof of Lemma 3.5.5 it was shown that for all $\rho > \rho^*_{ser}$, the approximation bound is never smaller than $c_\infty$. □

Combining Lemmas 3.5.1, 3.5.5, and 3.5.6 completes the proof of Theorem 7.

## 3.6 Serial Alignment: Lower Bound

In this section we present a result analogous to Theorem 6, by proving that the upper bound on the approximation ratio for serial alignment given in the previous section is tight in the limit as $\rho$ approaches $\infty$. Recall that $c_\infty = \lim_{\rho\to\infty} A_{ser}(\rho)$. (See Eq. (3.3) in Lemma 3.5.5.)

**Theorem 8** *For all sufficiently large $\rho$ there exists an input on which serial alignment achieves an approximation factor of at least $c_\infty + \frac{1}{27\rho^2}$*

The input configuration is structurally similar to the one used in the proof of Theorem 6, except for the choice of the angle $\alpha$. Here we select $\alpha$ such that $\sin\frac{\alpha}{2} = x_0/(2\rho)$, where $x_0$ is as defined in Eq. (3.2) in the proof of Lemma 3.5.5. Consider a fixed value $\rho > 2A_{ser}$. As in Section 3.4 we define the pattern point set $P = \{p_1, p_2, p_3, p_4\}$, where the first three points form an equilateral triangle of side length $2\rho$, and $p_4$ is placed at the midpoint of $\overline{p_1p_2}$. (See Fig. 12(a).) By an infinitesimal perturbation of the points, we may assume that the pair $(p_1, p_2)$ is the unique diametrical pair for $P$.

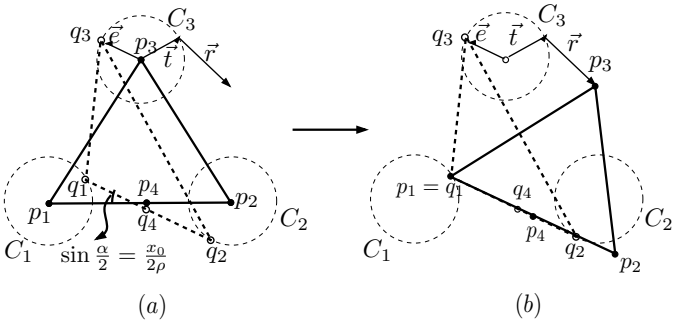


Fig. 12: The lower bound on $A_{ser}$.

We define the background set $Q = \{q_1, q_2, q_3, q_4\}$ so that the optimum Hausdorff distance will be at most 1. For $1 \leq i \leq 4$, let $C_i$ be a circle of unit radius centered $p_i$. Consider a lower tangent line at $C_2$ forming an angle $\alpha$ with line $\overline{p_1p_2}$. It is easy to verify that this line intersects $C_1$. Place $q_1$ at the rightmost intersection point of the tangent line and $C_1$, and place $q_2$ at the point of tangency with $C_2$. (See Fig. 12(a).) Running serial alignment so that $(p_1, p_2)$ is aligned with $(q_1, q_2)$ results in a translation and rotation. Let $\vec{t}$ and $\vec{r}$ denote the translation and rotational displacement vectors for $p_3$, respectively. Let $q_3 = p_3 + \vec{e}$, where $\vec{e}$ is a vector of unit length whose direction is chosen to be directly opposite that of $\vec{t} + \vec{r}$. Intuitively, $q_3$ has been chosen to be as far away as possible from $p_3$ after alignment. Finally, place $q_4$ at the midpoint of $\overline{q_1q_2}$. Observe that prior to alignment, each $p_i$ is within distance 1 from $q_i$, and therefore $h_{opt} \leq 1$.

Consider the placement of $P$ and $Q$ after running serial alignment. (See Fig. 12(b).) We will analyze the displacement distance of $p_3$ relative to $q_3$. As in the construction of Section 3.4, $p_4$ and $q_4$ were introduced to induce an asymmetry that forces the alignment of $(p_1, p_2)$ with $(q_1, q_2)$. We establish this formally in the next lemma.

**Lemma 3.6.1** *For all sufficiently large $\rho$, serial alignment will align $(p_1, p_2)$ with $(q_1, q_2)$.*

**Proof**: We reduce the proof to the analogous result for symmetric alignment, namely Lemma 3.4.1. In the proof of that lemma the only facts that we used were that $\rho$ is sufficiently large, and prior to alignment, each point of $P$ lies within

distance $h_{opt} \leq 1$ of the corresponding point of $Q$. These are both true in the present scenario.

Suppose that when we run serial alignment on the above input instance $(p_1, p_2)$ was not aligned with $(q_1, q_2)$. In Lemma 3.2.3 we showed that on all input instances, if serial alignment produces a Hausdorff distance of $d_{ser}$ then (without altering the individual point assignments) symmetric alignment will produce a Hausdorff distance of at most $d_{sym} \leq d_{ser} + h_{opt}$. In the proof of Lemma 3.4.1 it was argued that if $(p_1, p_2)$ was not aligned with $(q_1, q_2)$ in symmetric alignment, the resulting Hausdorff distance would be at least $\rho - h_{opt}$. Thus we have

$$\rho - h_{opt} \;\leq\; d_{sym} \;\leq\; d_{ser} + h_{opt},$$

which implies that $d_{ser} \geq \rho - 2h_{opt} \geq \rho - 2$. This distance will exceed $A_{ser}h_{opt}$ for all sufficiently large $\rho$, which yields the desired contradiction. □

To complete the analysis let us consider the translational and rotational displacement distances of $p_3$ that result from symmetric alignment. It is easy to see that $\vec{t} = \overrightarrow{p_1q_1}$, and so $\|\vec{t}\| = 1$. To determine the direction of $\vec{t}$, let us consider the points of $P$ prior to alignment. Let $\ell$ denote the line passing through $p_1$ that is parallel to $\overline{q_1q_2}$. Let $h_1$ and $h_2$ denote the orthogonal projections of $q_1$ and $q_2$ onto $\ell$, respectively. Since line $\overline{q_1q_2}$ is the lower tangent line of the circle $C_2$ at $q_2$, it follows that $p_2$, $q_2$, and $h_2$ are collinear. Combining the facts that $\|p_1p_2\| = 2\rho$, $\|p_2q_2\| = 1$, and $\ell$ is perpendicular to $\overline{p_2q_2}$, we have

$$\|q_2h_2\| \;=\; \|p_1p_2\| \sin\alpha - \|p_2q_2\| \;=\; 2\rho \sin\alpha - 1.$$

Because $\|q_1h_1\|$ is equal to $\|q_2h_2\|$, the angle between the line $\overline{p_1q_1}$ and $\ell$, denoted $\beta_t$, satisfies

$$\sin\beta_t \;=\; \frac{\|q_1h_1\|}{\|p_1q_1\|} \;=\; \frac{\|q_2h_2\|}{1} \;=\; 2\rho\sin\alpha - 1.$$

Since $\ell$ forms an angle of $\alpha$ with $\overline{p_1p_2}$ it follows that $\vec{t}$ forms an angle of $\phi_t = \beta_t - \alpha$, and hence expressing $\vec{t}$ in polar coordinates we have $\vec{t} = \langle 1, \phi_t\rangle$.

Next, let us consider the effect of rotation at $p_3$. Since $\triangle p_1p_2p_3$ is an equilateral triangle we have $\|p_1p_3\| = 2\rho$. By simple trigonometry (or see the comments at the start of the proof of Lemma 3.5.3) it follows that $p_3$'s displacement distance due to the rotation about $p_1$ by angle $\alpha$ is $2\rho\cdot 2\sin\frac{\alpha}{2} = 4\rho\sin\frac{\alpha}{2}$ and the displacement direction is

$$\left(\frac{\pi}{3} - \frac{\alpha}{2}\right) - \frac{\pi}{2} \;=\; -\frac{\pi}{6} - \frac{\alpha}{2}.$$

Let $\phi_r$ denote this angle. Thus, expressing $\vec{r}$ in polar coordinates we have

$$\vec{r} \;=\; \left\langle 4\rho\sin\frac{\alpha}{2}, \phi_r\right\rangle.$$

Let $D = \|\vec{t} + \vec{r}\|^2$ denote the length of the combined displacements. By the law of cosines we have

$$\begin{aligned} D &= \|\vec{t}\|^2 + \|\vec{r}\|^2 - 2\|\vec{t}\|\,\|\vec{r}\|\cos\left(\pi - (\phi_t - \phi_r)\right) \\ &= 1 + \left(4\rho\sin\frac{\alpha}{2}\right)^2 - 8\rho\sin\frac{\alpha}{2}\cos\left(\pi - (\beta_t - \alpha) + \left(-\frac{\pi}{6} - \frac{\alpha}{2}\right)\right) \\ &= 1 + \left(4\rho\sin\frac{\alpha}{2}\right)^2 + 8\rho\sin\frac{\alpha}{2}\sin\left(\frac{\pi}{3} + \frac{\alpha}{2} - \beta_t\right). \end{aligned}$$

Using the facts that $\sin\frac{\alpha}{2} = \frac{x_0}{2\rho}$ and $0 \le \alpha \le \pi/3$ and applying basic trigonometry we have

$$\begin{aligned} D &= 1 + (2x_0)^2 + 4x_0\left(\sin\left(\frac{\pi}{3}+\frac{\alpha}{2}\right)\cos\beta_t - \cos\left(\frac{\pi}{3}+\frac{\alpha}{2}\right)\sin\beta_t\right) \\ &\ge 1 + 4x_0^2 + 4x_0\left(\frac{\sqrt{3}}{2}\cos\beta_t - \frac{1}{2}\sin\beta_t\right). \end{aligned}$$

Note that $x_0$ is a constant slightly greater than $\sqrt{2/3}$. (See Eq. (3.2) in Lemma. 3.5.5.) To relate this to $c_\infty$ we begin with a variable substitution. Let

$$s = \cos\frac{\alpha}{2} = \sqrt{1-\left(\frac{x_0}{2\rho}\right)^2}.$$

Because $0 \le x_0 < 1$ and $\rho > \rho^*_{ser} > 1.5$ it follows directly that $3/4 < s \le 1$. We can restate a number of quantities in terms of $x_0$ and $s$.

$$\begin{aligned} \rho\sin\alpha &= 2\rho\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} = x_0 s, \\ \sin\beta_t &= 2\rho\sin\alpha - 1 = 2x_0 s - 1, \quad \text{and} \\ \cos\beta_t &= 2\sqrt{x_0 s - x_0^2 s^2}. \end{aligned}$$

Substituting these values yields

$$D \ge 1 + 4x_0^2 + 4x_0\left(\sqrt{3}\sqrt{x_0 s - x_0^2 s^2} - x_0 s + \frac{1}{2}\right).$$

By definition of $c_\infty$ we have $(c_\infty - 1)^2 = 1 + 2x_0 + 4\sqrt{3}x_0\sqrt{x_0 - x_0^2}$, and so it follows that

$$\begin{aligned} D &\ge 1 + 4x_0^2 + 4x_0\left(\sqrt{3}\sqrt{x_0 s - x_0^2 s^2} - x_0 s + \frac{1}{2}\right) + \\ &\qquad \left[(c_\infty - 1)^2 - \left(1 + 2x_0 + 4\sqrt{3}x_0\sqrt{x_0 - x_0^2}\right)\right] \\ &= (c_\infty - 1)^2 + 4x_0^2(1-s) + 4\sqrt{3}x_0\left(\sqrt{x_0 s - x_0^2 s^2} - \sqrt{x_0 - x_0^2}\right). \end{aligned}$$

It is straightforward to verify that $1/(2x_0) < s \leq 1$ for $\rho > \rho^*_{ser}$, and so it follows that $\sqrt{x_0 s - x_0^2 s^2} - \sqrt{x_0 - x_0^2} \geq 0$. We have

$$D \;\geq\; (c_\infty - 1)^2 + 4x_0^2(1-s).$$

Using the fact that $\sqrt{1-x} \leq 1 - x/2$ and $s = \sqrt{1 - \left(\frac{x_0}{2\rho}\right)^2}$ we have

$$D \;\geq\; (c_\infty - 1)^2 + 4x_0^2 \left[1 - \left(1 - \frac{1}{2}\left(\frac{x_0}{2\rho}\right)^2\right)\right] \;=\; (c_\infty - 1)^2 + \frac{x_0^4}{2\rho^2}.$$

Since $c_\infty < 3.2$ and $\sqrt{2/3} < x_0 < 1$ it is easy to verify that

$$D \;\geq\; (c_\infty - 1)^2 + \frac{x_0^4}{2\rho^2} \;\geq\; \left(c_\infty - 1 + \frac{x_0^4}{12\rho^2}\right)^2 \;\geq\; \left(c_\infty - 1 + \frac{1}{27\rho^2}\right)^2.$$

Therefore,

$$A_{ser}(\rho) \;\geq\; \frac{1 + \sqrt{D}}{h_{opt}} \;\geq\; c_\infty + \frac{1}{27\rho^2},$$

which completes the proof of Theorem 8.

## 3.7 Summary and Concluding Remarks

We have presented a simple modification to the alignment-based algorithm of Goodrich, Mitchell, and Orletsky [28]. Our modification has the same running time and retains the simplicity of the original algorithm. We have analyzed the approximation ratios for these algorithms as a function of the distance ratio $\rho$. We summarize and compare these approximation ratios in Fig. 13.

Note that the approximation ratio for symmetric alignment is smaller for all sufficiently large values of $\rho$. The cross-over point of two bounds is at $(\rho, A) \approx (1.54, 3.08)$. As a function of $\rho$, the highest upper bound for symmetric alignment

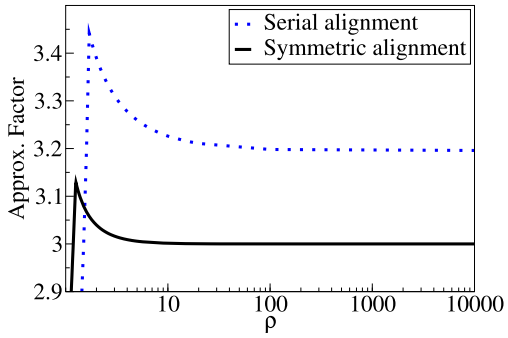


Fig. 13: The approximation ratios for serial alignment and symmetric alignment. (Note that the $y$-axis does not start at 0.)

is at $(\rho, A_{sym}) \approx (1.26, 3.14)$. The highest upper bound for serial alignment is at $(\rho, A_{ser}) \approx (1.72, 3.44)$. As $\rho$ approaches $\infty$, $A_{sym}(\rho)$ converges to 3 and $A_{ser}(\rho)$ converges to a value that is approximately 3.19. As mentioned earlier, in many applications of point pattern matching, large values of $\rho$ (exceeding 10, say) are to be expected. We have also shown that both bounds are nearly tight for large $\rho$, and they are tight in the limit.

# Chapter 4

# Embedding and Similarity Search for Point Sets under Translation

## 4.1 Introduction

In this chapter we are given a large database of point sets, which is to be preprocessed so that, given a query set, it is possible to efficiently compute its closest neighbor(s) in the database. We consider this problem in a relatively simple context, but one that still leads to quite an interesting computational problem. We assume that point sets have integer coordinates, that they are to be matched subject to an unknown translation, and that there is a significant fraction of outliers, that is, points from one set may not match any point of the other set. We assume, however, the when point coordinates match, they are identical (subject to the optimum translation). This is in contrast with measures such as the partial Hausdorff distance [35], where both outliers and near misses are tolerated. Outliers are challenging because global properties of the point sets, based for example on the identification of reference points such as centroids [2] are not applicable. The distance metric we use is the size of the symmetric difference of the two point sets, which is to be minimized

through some translation of one set relative to the other. (Formal definitions are given below.)

Our approach is based on finding an *embedding*, that is, a function that maps a point set into a metric space [37]. The distortion of such an embedding is defined to be the maximum multiplicative variation that distances might suffer in the mapping process. Our objective is to produce an embedding of low distortion, ideally into a space of low dimension, in which similarity search can be performed efficiently.

In the context of database search for point sets undergoing transformations, a well-known solution is based on geometric hashing [61]. This involves encoding the point positions of the points of each set relative to a small subset of points, called a *frame*, which uniquely determines the underlying transformation. (In the case of translation, a frame consists of a single point.) This approach is notably inefficient when a significant number of outliers are present. The reason stems from the fact that the points of the frame used to establish the matching transformation must be present in both sets. However, the presence of outliers confounds methods to select such a consistent frame from both sets. Suppose, for example, that there are $N$ possible frames to chose from, and the database encodes each point set using $k$ different frames. In the absence of some consistent method for selecting frames, the query will need to try at least $\Omega(N/k)$ frames, in order to achieve even a constant probability of succeeding in finding a common frame. Thus, irrespective of the value of $k$, the product of the space and query time will be suboptimal by a factor of $\Omega(N)$.

Another common approach, which is widely used in computer vision and image databases, is to compute some property that is invariant under motion. For example,

in image processing it is possible to construct a histogram of colors appearing in the image. The resulting invariant feature sets can be represented as vectors and can be compared using, for example, the Earth-Mover's distance (EMD) [3, 18]. The problem of computing low distortion embeddings of variants of EMD has been studied by Indyk and Thaper [40] and Wang, *et al.* [60]. Unfortunately, similarity of two color histograms provides no guarantees on the similarity of the underlying images.

The most popular invariant used for similarity search involving point sets is the *distance histogram*, which is defined to be the multiset of $\Theta(n^2)$ interpoint distances [9, 21]. Again, retrieval can be based on metrics like the EMD [13, 44]. Although the distance histogram and other invariant have been employed successfully in some applications, they do not provide a general solution, because similarity of such statistics does generally guarantee that the original objects match. For example, in the case of the distance histogram there exist so called *homometric sets* [50, 56], which have identical distance histograms, but which are not at all similar to one another. For example, the sets $P = \{0, 1, 4, 10, 12, 17\}$ and $Q = \{0, 1, 8, 11, 13, 17\}$ have identical distance histograms, but the symmetric difference of these sets under translation is never less than 6. Furthermore, there exist homometric sets whose symmetric difference under translation is arbitrarily large.

The principal contribution of this chapter is the first translation-invariant embedding for multi-dimensional point sets that provides provable quality guarantees. More formally, consider a point set consisting of at most $n$ points on the $d$-dimensional integer grid, where $d$ is a constant. We assume that each coordinate

of each point is bounded above by a polynomial function of $n$. Let $\mathbb{Z}$ denote the set of integers, and let $\mathbb{Z}_u$ denote $\{0, 1, 2, \ldots, u-1\}$. Let $\mathbb{Z}^d$ denote the set of $d$-element vectors over $\mathbb{Z}$, and define $\mathbb{Z}_u^d$ analogously for $\mathbb{Z}_u$. Finally, let $\mathbb{Z}_u^d(\leq n)$ denote the collection of point sets over $\mathbb{Z}_u^d$ that contain at most $n$ points. Given two finite sets $P$ and $Q$, let $P \ominus Q$ denote their *symmetric difference*, that is,

$$P \ominus Q \;=\; (P \setminus Q) \cup (Q \setminus P).$$

The cardinality of the symmetric difference is a well known metric on finite sets, which we denote by $|P \ominus Q|$.

It will be useful to extend these concepts to multisets as well. We define a *multiset* in $\mathbb{Z}_u^d(\leq n)$ to be one in which the total cardinality (counting multiplicities) is at most $n$. We generalize $|P \ominus Q|$ by summing the absolute differences of the multiplicities of corresponding elements. If $P$ and $Q$ are multisets of total cardinality at most $n$, then clearly $0 \leq |P \ominus Q| \leq 2n$.

Given a point set $P$ and any $\mathbf{t} \in \mathbb{Z}^d$, the *translate* $P + \mathbf{t}$ is defined to be $\{\mathbf{p} + \mathbf{t} \mid \mathbf{p} \in P\}$. Extending the symmetric difference, we define the *symmetric difference distance under translation*, denoted $\langle P \ominus Q \rangle$, to be

$$\langle P \ominus Q \rangle \;=\; \min_{\mathbf{t} \in \mathbb{Z}^d} |(P + \mathbf{t}) \ominus Q| \,.$$

For all of our results, the points $P$ and $Q$ will be assumed to be drawn from a finite domain $\mathbb{Z}_u^d$. It is understood that all arithmetic operations over individual coordinates are to be applied modulo $u$. This means that translation is subject to cyclic wraparound effects. It is easy to avoid these effects by first embedding the points in a region with a sufficiently large surrounding buffer. For example,

given $P, Q \subseteq \mathbb{Z}_u^d$, treat them as points in $\mathbb{Z}_{3u}^d$. Translate each point by the vector $(u, \ldots, u)$, thus mapping each set into the central third of $\mathbb{Z}_{3u}^d$ (see Fig. 1). In any optimal solution at least one point of $P$ coincides with a point of $Q$. It follows easily, that any solution that involves wraparound in $\mathbb{Z}_{3u}^d$ cannot be optimal.

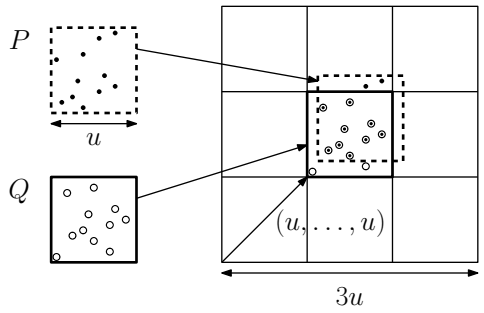


Fig. 1: Avoiding wraparound for point sets in $\mathbb{Z}_u^d$ by embedding them into $\mathbb{Z}_{3u}^d$.

Before presenting our main results, let us first define a few more terms. Let $\ell_1^d$ denote the metric space consisting of real $d$-dimensional space $\mathbb{R}^d$ endowed with the $L_1$ metric. Given $\mathbf{x}, \mathbf{y} \in \ell_1^d$, we denote their $L_1$ distance by $\|\mathbf{x} - \mathbf{y}\|_1 = \sum_i |x_i - y_i|$. We use the terms *randomized embedding* and *randomized function* throughout to denote an embedding or function computed by a randomized algorithm. Throughout, we use "log" to denote logarithm base 2 and "ln" to denote the natural logarithm.

Our first result states that there is a translation invariant randomized embedding of distortion $O(\log^2 n)$, that maps an $n$-element points set in $\mathbb{Z}^d$, with coordinates bounded by a polynomial in $n$, into $\ell_1^m$, where $m$ is roughly linear in $n$.

**Theorem 9** *Given sufficiently large integers $n$ and $u$, where $u \leq n^{O(1)}$, a constant $d$, and failure probability $\beta$, there exists a randomized embedding $\Psi \colon \mathbb{Z}_u^d(\leq n) \to \ell_1^m$, for $m = O\big(n \log^2 n \log \frac{1}{\beta}\big)$, such that for any two sets $P, Q \in \mathbb{Z}_u^d(\leq n)$:*

(*i*) $\|\Psi P - \Psi Q\|_1 \leq (2 \log n)\, \langle P \ominus Q \rangle$.

(*ii*) $\|\Psi P - \Psi Q\|_1 \geq \frac{1}{17 \log n}\, \langle P \ominus Q \rangle$, *with probability at least* $1 - \beta$, *and*

*This embedding can be computed in time* $O\big(n \log^4 n \log \frac{1}{\beta}\big)$.

Observe that part (i) holds unconditionally, irrespective of the algorithm's randomization. The probability that the distortion bound fails to hold for part (ii) can be made arbitrarily small, while adding a logarithmic factor to the dimension $m$.

The principal shortcoming of the above result is that the dimension of the space into which the points are embedded is superlinear in $n$, and hence is larger than the point set itself. Our other main result shows that the dimension can be reduced to a quantity that grows only logarithmically in $n$. The price that we pay is that the distortion bounds hold in expectation only, not with some given probability. The expected distortion is $O(\log^2 n)$.

**Theorem 10** *Given positive integers* $n$ *and* $u$, *where* $u \leq n^{O(1)}$, *and a constant* $d$, *there exists a randomized embedding* $\Psi' \colon \mathbb{Z}_u^d (\leq n) \to \ell_1^m$, *where* $m = O(\log n)$, *such that for any two sets* $P, Q \in \mathbb{Z}_u^d (\leq n)$:

(*i*) $\mathrm{E}\left[\|\Psi' P - \Psi' Q\|_1\right] \;\leq\; (3 \log n)\, \langle P \ominus Q \rangle$.

(*ii*) $\mathrm{E}\left[\|\Psi' P - \Psi' Q\|_1\right] \;\geq\; \frac{1}{17 \log n}\, \langle P \ominus Q \rangle$

*This embedding can be computed in expected time* $O(n \log^4 n)$.

We know of no prior work on this problem. In a 1-dimensional discrete setting, this problem is related to a version of edit distance on bit strings, where the number

of replacements corresponds to the symmetric difference in the sets, but it is possible to shift one set relative to the other without cost. The most closely related work to ours is that of Cormode and Muthukrishnan [20], on embedding strings under edit distance with moves. They present an embedding with distortion $O(\log n \log^* n)$ into $L_1$ with an exponential number of dimensions.

Here is a brief outline of our embedding algorithm. The basic embedding described in Theorem 9 involves a series of steps. First, we show that, through an appropriate projection, it is possible to reduce our problem to one involving 1-dimensional point sets of over $\mathbb{Z}_{u'}$, where $u' = n^{O(d)}$. We then reduce the size of the domain, by applying a translation-respecting randomized hash function, which maps the point set to $\mathbb{Z}_s$, where $s = O(n \log n)$. The resulting 1-dimensional integer point set can be viewed as a bit-vector in $\mathbb{Z}_2^s$. In order to obtain an embedding that is invariant under translations, we select various sized *probes*, each of which is a random subset of $\mathbb{Z}_s$. The application of a probe of size $\rho$ to a single placement of the vector produces an integer in $\mathbb{Z}_{2^\rho}$, where this integer is based on the bit pattern appearing in the bit-vector at each of the probed positions. We apply the probe at each of the $s$ positions (where indices are taken modulo $s$), and take the union of the probe results. Because probes are applied at every position, the result is invariant under translation. The result can be viewed as a vector in $\mathbb{Z}_{s+1}^{2^\rho}$. We then apply another hash function to reduce the dimension of this space from $\mathbb{Z}_{s+1}^{2^\rho}$ to $\mathbb{Z}_{s+1}^{O(s)}$.

Consider two points sets $P$ and $Q$. We show that, if $\rho$ is chosen in a manner that is sensitive to the actual distance between these point sets, then the distance between the two vectors resulting from this construction is related to the distance

between $P$ and $Q$ under translation. Since this distance is not known, we apply the construction to a series of exponentially increasing distance estimates, apply a suitable weight factor to each resulting vector, and then concatenate them together. In order to produce results that apply with the desired probability of success, we repeat the process some number of times, using different random hash functions and different random probes. We prove that the final result has the desired distortion properties.

In order to establish Theorem 10, we begin with the same approach as in the previous embedding, but we apply a sampling technique due to Kushilevitz, Ostrovsky, and Rabani [43] to further reduce the dimension. Recall that the above construction is applied multiple times with different distance estimates. We show that, if the distance estimate is sufficiently far from the optimum, then it is not necessary to use the entire vector and small random sample suffices. In order to produce the final embedding, we need to apply a weighting factor to each of these sampled vectors. Unfortunately, these weighting factors increases variances dramatically, and as a result, we are able to bound the distortion only in expectation.

In the next section we present a number of preliminary results. In Section 4.3, we present our translation invariant mapping. In Section 4.4, we show how to the reduce the dimension of the resulting mapping through random sampling. In Section 4.5, we present our embedding function and derive its distortion bounds. In Section 4.6, we discuss applications to similarity search for point sets stored in a database.

## 4.2 Preliminaries

Recall that a nonnegative function $d(P, Q)$ is a *metric* if (1) $d(P, P) = 0$, (2) $d(P, Q) = d(Q, P)$, and (3) $d(P, R) \leq d(P, Q) + d(Q, R)$. The last condition is the *triangle inequality*.

**Lemma 4.2.1** *The symmetric distance under translation is a metric.*

**Proof**: It is easy to see that the first two requirements of a metric hold. To establish the triangle inequality, let $T(P, Q, R) = \langle P \ominus Q \rangle + \langle Q \ominus R \rangle - \langle P \ominus R \rangle$. It suffices to show that $T(P, Q, R) \geq 0$. Let $t_1, t_2$, and $t_3$ denote the translations producing the minimal distance alignments for $\langle P \ominus Q \rangle$, $\langle Q \ominus R \rangle$, and $\langle P \ominus R \rangle$, respectively. Clearly, $|(P + t_3) \ominus R| \leq |(P + t_1) \ominus (R - t_2)|$ and $|(Q + t_2) \ominus R| = |Q \ominus (R - t_2)|$. Thus, we have

$$\begin{aligned} T(P, Q, R) &= |(P + t_1) \ominus Q| + |(Q + t_2) \ominus R| - |(P + t_3) \ominus R| \\ &\geq |(P + t_1) \ominus Q| + |Q \ominus (R - t_2)| - |(P + t_1) \ominus (R - t_2)| \\ &\geq 0, \end{aligned}$$

where the last implication follows from the triangle inequality for symmetric difference, which is well known to be a metric. □

Next, we observe that the problem of computing distances for point sets in the $d$-dimensional space $\mathbb{Z}_u^d$ under translation can be reduced to computing distances under translation in a 1-dimensional space $\mathbb{Z}_{O(u^d)}$. The result is based on the simple observation that we can unravel the $d$-dimensional grid into a sufficiently large 1-dimensional grid. Since the mapping is linear, it preserves similarity under trans-

lation. Here and throughout the chapter, when applying a function $g$ to a set $P$, we often denote the resulting set by $gP$.

**Lemma 4.2.2** *Consider a positive integer $u$ and constant $d$. There exists a function $g\colon \mathbb{Z}_u^d \to \mathbb{Z}_{u^d}$, such that for any sets $P, Q \subseteq \mathbb{Z}_u$, $\langle gP \ominus gQ\rangle = \langle P \ominus Q\rangle$. (Note that when computing $\langle gP \ominus gQ\rangle$, translation is performed in $\mathbb{Z}_{u^d}$, and when computing $\langle P \ominus Q\rangle$, translation is performed in $\mathbb{Z}_u^d$.) This function is computable in $O(1)$ time.*

**Proof**: Given $\mathbf{p} = (p_0, p_1, \ldots, p_{d-1}) \in \mathbb{Z}_u^d$, define $g(\mathbf{p}) = \sum_{i=0}^{d-1} p_i u^i$. Since we assume that $d$ is a constant, $g$ can be computed in $O(1)$ time.

We first observe that for $\mathbf{p}, \mathbf{q} \in \mathbb{Z}_u^d$, $\mathbf{p} = \mathbf{q}$ if and only if $g(\mathbf{p}) = g(\mathbf{q})$. To see this, suppose that $\mathbf{p} \neq \mathbf{q}$, and let $j$ be the largest index such that $p_j \neq q_j$. Without loss of generality we may assume that $p_j > q_j$, and so we have

$$\begin{aligned} g(\mathbf{p}) - g(\mathbf{q}) &= \left(\sum_{i=0}^{j-1}(p_i - q_i)u^i\right) + (p_j - q_j)u^j \;\geq\; \left(\sum_{i=0}^{j-1}(1-u)u^i\right) + 1 \cdot u^j \\ &= (1-u)\frac{u^j - 1}{u-1} + u^j \;>\; 0. \end{aligned}$$

Therefore, $g$ is a 1–1 function from $\mathbb{Z}_u^d$ to $\mathbb{Z}_{u^d}$. Since these sets have equal cardinalities, this is a bijection. Therefore, $|P \ominus Q| = |gP \ominus gQ|$.

Observe that, since $g$ is a linear function of the point coordinates, $g(\mathbf{p} + \mathbf{t}) = g(\mathbf{p}) + g(\mathbf{t})$. Let $\mathbf{t} \in \mathbb{Z}_u^d$ denote the optimal $d$-dimensional translation, so that $\langle P \ominus Q\rangle = |(P + \mathbf{t}) \ominus Q|$. Let $s \in \mathbb{Z}_{u^d}$ be the optimal 1-dimensional translation so that $\langle gP \ominus gQ\rangle = |(gP + s) \ominus gQ|$. Applying the observations above, we have

$$\begin{aligned} \langle P \ominus Q\rangle &= |(P + \mathbf{t}) \ominus Q| \;\leq\; \left|(P + g^{-1}(s)) \ominus Q\right| \\ &= \left|g(P + g^{-1}(s)) \ominus gQ\right| \;=\; |(gP + s) \ominus gQ| \;=\; \langle gP \ominus gQ\rangle\,. \end{aligned}$$

A symmetrical argument, applied $gP$ and $gQ$ and $s$, implies that $\langle gP \ominus gQ \rangle \leq \langle P \ominus Q \rangle$. Therefore, $\langle gP \ominus gQ \rangle = \langle P \ominus Q \rangle$, as desired. □

Next, we present a couple of utility results. The first is a straightforward observation that the result of applying a randomized hash function with a low collision probability to each of the elements of a pair of multisets produces two sets whose symmetric difference is similar to the original sets. (Recall that the symmetric difference distance is defined on multisets by counting not just the number of mismatched elements, but the absolute differences in the multiplicities of each distinct element.)

**Lemma 4.2.3** *Let $0 \leq \gamma \leq 1$, and suppose that we are given a randomized function $h\colon \mathbb{Z} \to \mathbb{Z}$ such that for all distinct $x, y \in \mathbb{Z}$, $\Pr[h(x) = h(y)] \leq \gamma$. Given a positive integer $n$ and multisets $P, Q \in \mathbb{Z}(\leq n)$, let $hP$ and $hQ$ denote the sets (not multisets) that result by applying $h$ to each element of $P$ and $Q$, respectively. Given any (failure probability) $0 < \beta \leq 1$,*

*(i)* $|hP \ominus hQ| \leq |P \ominus Q|$, *and*

*(ii)* $|hP \ominus hQ| \geq \left(1 - \frac{2n\gamma}{\beta}\right) |P \ominus Q|$, *with probability at least* $1 - \beta$.

**Proof**: Part (i) is trivial, since applying any given function to both $P$ and $Q$ can only decrease the size of the symmetric difference. To prove (ii), let $\delta = |P \ominus Q|$. Define a *collision* to be any distinct pair $x, y \in \mathbb{Z}$, such that $h(x) = h(y)$. Observe that this collision can affect the size of the symmetric difference only if both $x$ and $y$ are in $P \cup Q$, and at least one is in $P \ominus Q$.

Let $M$ denote the distinct elements in $P \cup Q$, and let $m = |M|$. Clearly, $m \leq 2n$. Consider any fixed element $x$ in $M$. Let $\delta_x$ denote the absolute difference

between the multiplicities of $x$ in $P$ and $Q$, and hence $\delta = \sum_x \delta_x$. For $x, y \in P \cup Q$, define $I_{x,y}$ to be an indicator random variable whose value is 1 if $h(x) = h(y)$ and 0 otherwise. Let

$$C_x \;=\; \sum_{y \in M \setminus \{x\}} \delta_x \, I_{x,y}.$$

The symmetric difference decreases only when an element of $P \ominus Q$ collides with some other element of $P \cup Q$. Thus, it is easy to see that $|P \ominus Q| - |hP \ominus hQ| \;\leq\; \sum_{x \in P \ominus Q} C_x$. We also have

$$\mathrm{E}\left[C_x\right] \;=\; \sum_{y \in M \setminus \{x\}} \delta_x \Pr\left[h(x) = h(y)\right] \;<\; m \delta_x \gamma.$$

Thus,

$$\begin{aligned} \mathrm{E}\left[\, |P \ominus Q| - |hP \ominus hQ| \,\right] \;&\leq\; \mathrm{E}\left[\sum_{x \in P \ominus Q} C_x\right] \;=\; \sum_{x \in P \ominus Q} \mathrm{E}\left[C_x\right] \\ &<\; m\gamma \sum_{x \in P \ominus Q} \delta_x \;=\; m\gamma\delta \;\leq\; 2n\gamma\delta. \end{aligned}$$

By Markov's inequality, $\Pr\left[|P \ominus Q| - |hP \ominus hQ| \geq 2n\gamma\delta/\beta\right] \;\leq\; \beta$. Recalling that $\delta = |P \ominus Q|$, it follows that with probability at least $(1 - \beta)$, we have

$$|hP \ominus hQ| \;\geq\; \left(1 - \frac{2n\gamma}{\beta}\right) |P \ominus Q| \,,$$

as desired. □

Our second utility lemma will be useful for compressing space. For positive integers $u$ and $s$, consider hash function $h : \mathbb{Z}_u \to \mathbb{Z}_s$. We say that $h$ is *translation respecting* if, for any $x$ and $t$, $h(x+t) = h(x) + h(t)$. (Observe that $x+t$ is computed over $\mathbb{Z}_u$, and $h(x)+h(t)$ is computed over $\mathbb{Z}_s$.) Common choices for low-collision hash functions, such as the universal hash function $h(x) = ((ax + b \mod u) \mod s)$, do

not satisfy this condition. Our next lemma presents such a function for the domain of values in interest. It is similar to an approach that has used in the context of string pattern matching [42].

**Lemma 4.2.4** *Consider positive integers $n$ and $u$, where $u \leq n^c$ for some constant $c \geq 1$. For any $\alpha$ and $\beta$, where $0 \leq \alpha, \beta \leq 1$, there exists $s = \Theta((n \log n)/(\alpha\beta))$ (with constant factors depending on $c$) and a randomized translation-respecting hash function $h\colon \mathbb{Z}_u \to \mathbb{Z}_s$, such that for any sets $P, Q \in \mathbb{Z}_u (\leq n)$ we have*

*(i)* $|hP \ominus hQ| \leq |P \ominus Q|$, *and*

*(ii)* $|hP \ominus hQ| \geq (1-\alpha)\,|P \ominus Q|$, *with probability at least* $1-\beta$.

*The function $h$ is computable in $O(1)$ time (assuming that arithmetic operations on numbers of magnitude at most $s$ can be computed in constant time).*

**Proof**: As before, (i) is trivial, and so we concentrate on proving (ii). Let $\sigma = 6 + (8c/(\alpha\beta))$ and let $r = \sigma n \ln(\sigma n)$. Let $R$ denote the set of prime numbers in the range $n \ln n$ to $r$. By the Prime Number Theorem [51], for all sufficiently large $n$ the number of primes less than or equal to $n$ is at least $n/(2 \ln n)$ and at most $3n/(2 \ln n)$. (If $n$ is so small that this does not hold, we simply set $s = u$ and define $h$ to be the identity function.) It follows that for all sufficiently large $n$ we have

$$|R| \;\geq\; \frac{\sigma n \ln(\sigma n)}{2 \ln(\sigma n \ln(\sigma n))} - \frac{3n \ln n}{2 \ln(n \ln n)} \;\geq\; \frac{\sigma n \ln(\sigma n)}{4 \ln(\sigma n)} - \frac{3n}{2} \;\geq\; \left(\frac{\sigma - 6}{4}\right) n = \frac{2c}{\alpha\beta} n.$$

For any $s \in R$, define $h_s(x) = x \bmod s$. Linearity follows since

$$h_s(x+t) \;=\; (x+t) \bmod s \;=\; (x \bmod s) + (t \bmod s) \;=\; h_s(x) + h_s(t).$$

(The addition $x + t$ is performed over $\mathbb{Z}_u$, and all others are performed over $\mathbb{Z}_s$.)

For any distinct $x, y \in \mathbb{Z}_u$, $h_s(x) = h_s(y)$ if and only if $|x - y|$ is divisible by $s$. Since $x, y \leq u \leq n^c$, and $s \geq n$, it follows that this can be true for at most $c$ choices of $s$. We define $h(x)$ to be $h_s(x)$, where $s$ is a random element of $R$. Observe that for any fixed $x, y \in \mathbb{Z}_u$,

$$\Pr\left[h(x) = h(y)\right] \;\leq\; \frac{c}{|R|} \;=\; \frac{\alpha\beta}{2n}.$$

By applying Lemma 4.2.3(ii) with $\gamma = \alpha\beta/(2n)$ and failure probability $\beta$, it follows that

$$|hP \ominus hQ| \;\geq\; \left(1 - \frac{2n\gamma}{\beta}\right)|P \ominus Q| \;=\; (1-\alpha)\,|P \ominus Q|$$

holds with probability at least $(1 - \beta)$, as desired. □

Henceforth, we may assume that $u \geq s$, since otherwise we may set $u = s$ and take $h$ to be the identity function.

**Lemma 4.2.5** *The bounds of Lemma 4.2.4 apply to the symmetric difference distance under translation. That is,*

(*i*) $\langle hP \ominus hQ\rangle \;\leq\; \langle P \ominus Q\rangle$, *and*

(*ii*) $\langle hP \ominus hQ\rangle \;\geq\; (1-\alpha)\,\langle P \ominus Q\rangle$, *with probability at least* $1 - \beta$.

**Proof**: Let $s$ and $h = h_s$ be as defined in Lemma 4.2.4. For each $t \in \mathbb{Z}_s$, $h(t) = (t \bmod s) = t$, and hence by our assumption that $u \geq s$, for each $t' \in \mathbb{Z}_s$, there exists $t \in \mathbb{Z}_u$ such that $t' = h(t)$. Combining this with the the fact that $h$ is translation

respecting and Lemma 4.2.4(i) we have

$$\begin{aligned}\langle hP \ominus hQ\rangle &= \min_{t'\in\mathbb{Z}_s} |(hP + t') \ominus hQ| = \min_{t\in\mathbb{Z}_u} |(hP + h(t)) \ominus hQ| \\ &= \min_{t\in\mathbb{Z}_u} |(h(P + t) \ominus hQ| \leq \min_{t\in\mathbb{Z}_u} |(P + t) \ominus Q| = \langle P \ominus Q\rangle .\end{aligned}$$

Part (ii) follows analogously by applying Lemma 4.2.4(ii) to the sets $P + t$ and $Q$, where $t$ is the optimal aligning transformation:

$$\langle hP \ominus hQ\rangle = \min_{t\in\mathbb{Z}_u} |(h(P + t) \ominus hQ| \geq \min_{t\in\mathbb{Z}_u}(1-\alpha) |(P + t) \ominus Q| = (1-\alpha) \langle hP \ominus hQ\rangle ,$$

with probability at least $1 - \beta$. □

The following lemma is similar to the previous one, but it will be applied in a context where the translation-respecting property of the hash function is not required. This additional flexibility allows us to remove the restriction in the size of $u$. It follows by applying Lemma 4.2.3 to a suitable universal hash function [12]. The choice of the hash function given in the proof will be discussed later when we consider execution times. This lemma applies more generally to multisets.

**Lemma 4.2.6** *Consider positive integers $n$ and $u$. For all $\alpha$ and $\beta$, where $0 \leq \alpha, \beta \leq 1$, there exists $s = O(n/(\alpha\beta))$ and a randomized function $h\colon \mathbb{Z}_u \to \mathbb{Z}_s$ such that for any multisets $P, Q \in \mathbb{Z}_u(\leq n)$ the multisets $hP$ and $hQ$ satisfy:*

*(i)* $|hP \ominus hQ| \leq |P \ominus Q|$, *and*

*(ii)* $|hP \ominus hQ| \geq (1 - \alpha) |P \ominus Q|$, *with probability at least* $1 - \beta$.

*We may further assume that $s$ is a power of 2.*

**Proof**: Let $s$ be the smallest power of 2 that is at least as large as $2n/(\alpha\beta)$. For the purposes of defining the function, it will be convenient to express each element of $\mathbb{Z}_u$ (resp., $\mathbb{Z}_s$) as a bit vector of length $\lceil \log u \rceil$ (resp., $\log s$). Thus, $P$ and $Q$ can be viewed as multisets of cardinality at most $n$ where elements are drawn from $\mathbb{Z}_2^{\lceil \log u \rceil}$.

Given a matrix $M \in \{0,1\}^{(\log s) \times \lceil \log u \rceil}$ and a vector $b \in \mathbb{Z}_2^{\log s}$, define $h_{M,b} \colon \mathbb{Z}_2^{\lceil \log u \rceil} \rightarrow \mathbb{Z}_2^{\log s}$ to be

$$h_{M,b}(x) \;=\; Mx + b,$$

where arithmetic operations are performed over $\mathbb{Z}_2$. It is well known that if $M$ and $b$ are randomly generated, the resulting randomized function $h = h_{M,b}$ is a universal hash function [12,27], and for any fixed $x, y \in \mathbb{Z}_u$, $\Pr[h(x) = h(y)] \leq 1/s \leq \alpha\beta/(2n)$.

Given this bound on the collision probability, the rest is structurally similar to the proof of Lemma 4.2.4. In particular, (i) is trivial, and to show (ii) we apply Lemma 4.2.3(ii) with $\gamma = \alpha\beta/(2n)$ and failure probability $\beta$. We have

$$|hP \ominus hQ| \;\geq\; \left(1 - \frac{2n\gamma}{\beta}\right) |P \ominus Q| \;\geq\; (1-\alpha)\,|P \ominus Q|\,,$$

with probability at least $(1-\beta)$, as desired. □

## 4.3 Translation-Invariant Mapping

The purpose of this section is to present a randomized function that maps a point set in $\mathbb{Z}_s (\leq n)$ to an integer vector, such that this function is invariant under translation. We will then show that, by applying weighted repetitions of this function, it is possible to produce the randomized embedding function described in Theorem 9.

Before presenting this transformation, consider a point set $P' \in \mathbb{Z}_s (\leq n)$. (In our algorithm, this set will arise as the image of $P$ under a hash function that maps elements to $\mathbb{Z}_s$.) Observe that we may interpret $P'$ as a bit-vector in an $s$-dimensional space, in particular, as an element of $\mathbb{Z}_{2^s}$. More precisely, we represent $P'$ as $[p_0, \ldots, p_{s-1}]$, where $p_i = 1$ if $i \in P'$ and 0 otherwise. Given any translation $t \in \mathbb{Z}_s$, we will continue to use the notation $P' + t$ to denote the translation of $P'$ by $t$ modulo $s$, which in this context corresponds to a right circular shift of the elements of this bit vector by $t$ positions.

Given a positive integer $\rho$, define an $(s, \rho)$-*probe* to be a $\rho$-element vector $\pi = (i_1, i_2, \ldots, i_\rho)$, where $i_j \in \mathbb{Z}_s$, for $1 \leq j \leq \rho$. We say that such a probe is *random* if each element $i_j$ is sampled independently at random from $\mathbb{Z}_s$ with replacement. Given $P' \in \mathbb{Z}_2^s$, define $P'[\pi]$ to be the integer whose bit representation is $\langle p_{i_1} p_{i_2} \ldots p_{i_\rho} \rangle$, and define the multiset

$$\widehat{\Phi}_\pi P' \;=\; \{(P' + t)[\pi] : t \in \mathbb{Z}_s\}\,.$$

Observe that this is a multiset because different translations may generate the same bit pattern. The total cardinality of $\widehat{\Phi}_\pi P$ (counting multiplicities) is $s$, and its elements are drawn from $\mathbb{Z}_{2^\rho}$. Because the probe is applied uniformly to all translations in $\mathbb{Z}_s$ we have:

**Lemma 4.3.1** *Given any $(s, \rho)$-probe $\pi$, $\widehat{\Phi}_\pi$ is invariant under translation. That is, for all $t \in \mathbb{Z}_s$, $\widehat{\Phi}_\pi(P + t) = \widehat{\Phi}_\pi P$.*

A simple 5-point example of this transformation is shown in Figure 2, where we have chosen $s = 11$ and $\rho = 4$. First, we hash the elements of $P$ into $\mathbb{Z}_{11}$ by

the hash function $h'(x) = x \bmod s$. Let $P' = h'P$ be the resulting set, which we interpret as a binary vector in $\mathbb{Z}^{11}$. Let $\pi = [0, 3, 6, 7]$ be our random probe vector. For $t = 0$, we have $(P' + t)[\pi] = [P'[0], P'[3], P'[6], P'[7]] = [1, 1, 1, 0]$, which can be represented as the bit vector $\langle 1110 \rangle$. By repeatedly shifting this probe pattern (circularly) through $P'$, we generate the multiset of bit vectors that form $\widehat{\Phi}_\pi P'$ as shown in Figure 2. We may interpret this multiset as a vector in $\mathbb{Z}^{2^\rho} = \mathbb{Z}^{16}$, where the $i$th entry of this vector is the number of occurrences of the bit vector whose integer value is $i$. This is shown in the last line of Figure 2. Since $2^\rho$ may be quite large, and this vector contains at most $s$ nonzero entries, we will ultimately compress its length through a second hash function, denoted $h''$, which will be described later. In this simple example we have taken $h''$ to be the identity.

| Object | Notes | Example |
|---|---|---|
| $P \in \mathbb{Z}_u (\leq n)$ | $u = 24$ | $\{3, 6, 10, 14, 22\}$ |
| $P' \in \mathbb{Z}_s (\leq n)$ | $s = 11$ | 0 1 2 3 4 5 6 7 8 9 10 / 1 0 0 1 0 0 1 0 0 0 1 |
| $\widehat{\Phi}_\pi P' \subseteq \mathbb{Z}_{2^\rho} (\leq s)$ | $\left\lvert\widehat{\Phi}_\pi\right\rvert = s$ | $\{1110, 0000, 0000, 1101, 0011, 0010, 1000, 0101, 0110, 0000, 1001\}$ |
| $h''\widehat{\Phi}_\pi P'$ | | 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111 / 3 0 1 1 0 1 1 0 1 1 0 0 0 1 1 0 |

Fig. 2: An example of the invariant transformation, where $n = 5$, $u = 24$, $s = 11$, $h'(x) = x \bmod s$, and $\pi = [0, 3, 6, 7]$. (For simplicity we have chosen the second hash function $h''$ to be the identity.)

Given two sets $P'$ and $Q'$, let $\delta^* = \langle P' \ominus Q' \rangle$ denote their distance under translation. Our next lemma shows that the distance between the transformed objects can be bounded by a function of $\delta^*$ and the probe length $\rho$. Intuitively, as $\rho$ increases, the likelihood of encountering a mismatch increases (depending on the distance between the point sets). Part (i) of the lemma asserts that the $L_1$ distance between the resulting vectors grows at most linearly in each of $\rho$ and $\delta^*$. Part (ii) asserts that if $\rho$ is sufficiently large, the probability of encountering a mismatch is so high that the distance between the resulting vectors will be almost as high as the maximum possible value of $2s$.

**Lemma 4.3.2** *Consider a positive integer $s$ and two sets $P', Q' \in \mathbb{Z}_s(\leq n)$, and let $\delta^* = \langle P' \ominus Q' \rangle$. Given a positive integer $\rho$, let $\pi$ be a random $(s, \rho)$-probe. Then there exists a translation-invariant function $\widehat{\Phi}_\pi : \mathbb{Z}_s(\leq n) \to \mathbb{Z}_{2^\rho}(\leq s)$, such that*

(*i*) $\left\| \widehat{\Phi}_\pi P' - \widehat{\Phi}_\pi Q' \right\|_1 \leq 2\rho\delta^*$, *and*

(*ii*) *if* $\rho \geq (3s \ln s)/\delta^*$ *then* $\left\| \widehat{\Phi}_\pi P' - \widehat{\Phi}_\pi Q' \right\|_1 \geq 2s - 2$ *with probability at least* $\left(1 - \frac{2}{s}\right)$.

**Proof**: Let $\widehat{\Phi}_\pi$ be the above probing function, which by Lemma 4.3.1 is translation invariant. Since $\delta^* = \langle P' \ominus Q' \rangle$, this means that for some translation of $P'$ there are $\delta^*$ elements of $\mathbb{Z}_s$ in the symmetric difference $P' \ominus Q'$. Because $\widehat{\Phi}$ is invariant under translations, by applying a suitable translation to $P'$, there is no loss in generality if we assume that this optimal translation is the identity.

Since $\pi$ has $\rho$ elements, there are at most $\rho\delta^*$ choices of translations $t$ such that the result of $(P' + t)[\pi]$ accesses one of these mismatched elements. Each of these

may produce a probe value that fails to match any of the probe results of $\widehat{\Phi}_\pi Q'$. All the others placements will match the corresponding probe of $Q'$. Symmetrically, there are $\rho\delta^*$ choices of $r$ such that $(Q'+r)[\pi]$ fails to match any probe value of $\widehat{\Phi}_\pi P'$, but all others will match. Thus we have

$$\left\| \widehat{\Phi}_\pi P' - \widehat{\Phi}_\pi Q' \right\|_1 \;\leq\; 2\rho\delta^*,$$

which establishes (i).

In order to establish (ii), given any pair of translations $t, r \in \mathbb{Z}_s$, let $\delta_{t,r} = |(P'+t) \ominus (Q'+r)|$. Clearly, for any $t, r \in \mathbb{Z}_s$, $\delta_{t,r} \geq \delta^*$. For any particular probe placement $\pi$, the probe values $(P'+t)[\pi]$ and $(Q'+r)[\pi]$ match if and only if all of the probed positions match. Since the indices of $\pi$ are chosen at random from $\mathbb{Z}_s$, each position matches with probability $(1-\delta_{t,r}/s)$. Since the elements of $\pi$ are chosen independently at random, we have

$$\Pr\left[(P'+t)[\pi] = (Q'+r)[\pi]\right] \;=\; \left(1-\frac{\delta_{t,r}}{s}\right)^\rho \;\leq\; \left(1-\frac{\delta^*}{s}\right)^\rho \;\leq\; e^{-\delta^*\rho/s}.$$

By the union bound we have

$$\Pr\left[(P'+t)[\pi] \in \widehat{\Phi}_\pi Q'\right] \;=\; \bigcup_{r\in\mathbb{Z}_s} \Pr\left[(P'+t)[\pi] = (Q'+r)[\pi]\right] \;\leq\; \sum_{r\in\mathbb{Z}_s} e^{-\delta^*\rho/s} \;=\; s e^{-\delta^*\rho/s}.$$

Let $X_{P'}$ be a random variable whose value is the number of probe placements $t$ such that $(P'+t)[\pi] \in \widehat{\Phi}_\pi Q'$. By the linearity of expectation and since $\rho \geq (3s\ln s)/\delta^*$, we have

$$\mathrm{E}\left[X_{P'}\right] \;=\; \sum_{t\in\mathbb{Z}_s} \Pr\left[(P'+t)[\pi] \in \widehat{\Phi}_\pi Q'\right] \;\leq\; s^2 e^{-\delta^*\rho/s} \;\leq\; s^2 e^{-(3s\ln s)/s} \;=\; \frac{s^2}{s^3} \;=\; \frac{1}{s}.$$

By Markov's inequality, it follows that $\Pr\left[X_{P'} \geq 1\right] \leq \frac{1}{s}$. By defining $X_{Q'}$ symmetrically, it follows that $\Pr\left[X_{Q'} \geq 1\right] \leq \frac{1}{s}$. Since $\widehat{\Phi}_\pi P'$ and $\widehat{\Phi}_\pi Q'$ each contain $s$ values

(including multiplicities), their $L_1$ distance is $2s$ minus the number of matches. And so with probability at least $\left(1 - \frac{2}{s}\right)$ we have

$$\left\| \widehat{\Phi}_\pi P' - \widehat{\Phi}_\pi Q' \right\|_1 \;=\; 2s - (X_{P'} + X_{Q'}) \;\geq\; 2s - 2,$$

as desired. □

Using the above lemma, we present the main utility result upon which our translation-invariant transformation is based. This lemma shows that, given two points sets over $\mathbb{Z}_u$ each containing at most $n$ elements and an estimate $\delta$ on their distance, there exists a function $\Phi$ that maps each point set to a vector such that the distance between the two resulting vectors reveals whether $\delta$ is significantly greater or significantly smaller than the optimal distance $\delta^*$. Intuitively, this is done by randomly probing the point sets such that, under the assumption that $\delta = \delta^*$, the probability of encountering a mismatch will be a constant. Thus, if $\delta$ is either significantly larger or significantly smaller than $\delta^*$, the distance between the resulting vectors will reflect this. (In subsequent sections, we will explore different ways of dispensing with this distance estimate.)

As mentioned earlier, the process involves three stages: first, a randomized translation-respecting hash function is invoked to reduce the domain size to $O(n \log n)$, next, a random probe sequence is applied to produce a sparse translation-invariant vector of very large size, and finally, a second hash function is invoked to reduce the size of this vector. We start with a weak form of the process, which achieves the desired goals with constant probability. (Later, we shall show how to boost the probability of success through repeated trials.)

**Lemma 4.3.3** *Consider positive integers $n$ and $u$, where $u \leq n^c$ for some constant $c \geq 1$, and a distance estimate $1 \leq \delta \leq 2n$. There exists a randomized function $\Phi'' \colon \mathbb{Z}_u \to \mathbb{Z}_{s+1}^m$, where $s = \Theta(n \log n)$ and $m = O(n \log n)$, that satisfies the following property. Consider any two sets $P, Q \subseteq \mathbb{Z}_u$, each of cardinality at most $n$, and let $\delta^*$ denote $\langle P \ominus Q \rangle$. Then*

*(i)* $\|\Phi'' P - \Phi'' Q\|_1 \leq s\frac{\delta^*}{\delta}$, *and*

*(ii) if* $\delta \leq \frac{\delta^*}{8 \ln s}$ *then* $\|\Phi'' P - \Phi'' Q\|_1 \geq \frac{3s}{2}$, *with constant probability.*

*More precisely, given any $0 < \beta_0 < 1$, case (ii) holds with probability at least $(1 - \beta_0)^3$. For any $P \in \mathbb{Z}_u$, $\|\Phi'' P\|_1 \leq s$.*

Before presenting the proof, let us explore the utility of this lemma. Observe first that, if $\delta > \frac{2\delta^*}{3}$, then by part (i) it follows that $\|\Phi'' P - \Phi'' Q\|_1 < \frac{3s}{2}$. On the other hand, if $\delta < \frac{\delta^*}{8 \ln s}$, then by part (ii) this inequality will fail to hold with constant probability. Thus, except for the interval $\delta \in \delta^* \left[\frac{1}{8 \ln s}, \frac{2}{3}\right]$, this test will allow us to determine whether the estimate $\delta$ is larger or smaller than $\delta^*$, with constant probability. The distortion in our final embedding arises directly as a consequence of this interval of uncertainty.

**Proof**: (of Lemma 4.3.3)

By Lemmas 4.2.4(i) and 4.2.6(i), the functions that they produce cannot increase distances. By Lemma 4.3.2 therefore, we have

$$\|\Phi'' P - \Phi'' Q\|_1 \;\leq\; 2\rho\delta^* \;\leq\; 2\frac{s}{2\delta}\delta^* \;\leq\; \frac{s\delta^*}{\delta},$$

which establishes (i).

To establish (ii), let $\alpha_0 = 1/16$ and $\beta_0 = 1/64$. We begin by applying Lemma 4.2.4, with $\alpha = \alpha_0$ and $\beta = \beta_0$. Let $h' \colon \mathbb{Z}_u \to \mathbb{Z}_s$ denote the resulting function, where $s = \Theta(n \log n)$. We may assume that $s \geq 128$. If we let $P' = h'P$ and $Q' = h'Q$, by Lemma 4.2.5(ii), with probability at least $1 - \beta_0$, we have

$$\langle P' \ominus Q' \rangle \;\geq\; (1 - \alpha_0) \langle P \ominus Q \rangle \;=\; (1 - \alpha_0)\delta^*.$$

Let $\rho = \lfloor s/(2\delta) \rfloor$. Since $\delta^* \leq 2n \leq 2s$ and by our hypotheses that $\delta \leq \delta^*/(8 \ln s)$ and $s \geq 128$, it is easy to show that $\rho \geq (3s \ln s)/\delta^*$. Therefore, we may apply Lemma 4.3.2(ii) to the sets $P'$ and $Q'$. Let $\widehat{\Phi}_\pi$ denote the resulting translation-invariant transformation, and let $\widehat{P} = \widehat{\Phi}_\pi P'$ and $\widehat{Q} = \widehat{\Phi}_\pi Q'$. We have $\left\| \widehat{P} - \widehat{Q} \right\|_1 \geq 2s - 2$, with probability at least $(1 - \frac{2}{s})$.

$\widehat{P}$ and $\widehat{Q}$ are each multisets of cardinality $s$ over $\mathbb{Z}_{2^\rho}$, and hence each can be interpreted as an integer vector in $\mathbb{R}^{2^\rho}$, in which the $i$th component is the number of occurrences of element $i$. Under this interpretation, the (multiset) symmetric difference distance is equivalent to the $L_1$ distance between $\widehat{P}$ and $\widehat{Q}$. Applying these two different interpretations of $\widehat{P}$ and $\widehat{Q}$, we have

$$\left| \widehat{P} \ominus \widehat{Q} \right| \;=\; \left\| \widehat{P} - \widehat{Q} \right\|_1 .$$

By the definition of $\rho$ we have $\rho + 1 \geq s/(2\delta)$, and thus $\delta \geq s/(2(\rho+1))$. Since $s = \Theta(n \log n)$ and $\delta \leq 2n$, it follows that $\rho = \Omega(\log n)$. We may assume therefore that $n$ is sufficiently large that $\rho \geq 6$, from which it follows that $7/(2(\rho + 1)) \geq \frac{3}{\rho}$. Combining all of this, we find that, with probability at least $1 - \beta_0$,

$$\langle P' \ominus Q' \rangle \;\geq\; (1 - \alpha_0)\delta^* \;>\; \frac{7}{8}\delta^* \;\geq\; 7\delta \ln s \;\geq\; \frac{7s \ln s}{2(\rho + 1)} \;\geq\; \frac{3s \ln s}{\rho}.$$

Therefore, we may apply Lemma 4.3.2(ii) to obtain that, with probability at least $(1-\beta_0)\left(1-\frac{2}{s}\right)$,

$$\left\|\widehat{P}-\widehat{Q}\right\|_1 \;\geq\; 2s-2.$$

Given our assumption that $s \geq 128$ and $\beta_0 = 1/64$, the probability of this holding is at least $(1-\beta_0)^2$.

Finally, we apply Lemma 4.2.6 to the multisets $\widehat{P}$ and $\widehat{Q}$, where $n = s$, $u = 2^\rho$, $\alpha = \alpha_0$ and $\beta = \beta_0$. Let the resulting function be $h'' \colon \mathbb{Z}_{2^\rho} \to \mathbb{Z}_{O(s)}$, and let $P'' = h''\widehat{P}$ and $Q'' = h''\widehat{Q}$. (Given that $\rho$ may be as large as $\Theta(n \log n)$, we will not compute this function explicitly. Computational issues will be discussed later.) Thus, we have

$$\begin{aligned}\|P''-Q''\|_1 &= \left|h''\widehat{P} \ominus h''\widehat{Q}\right| \;\geq\; (1-\alpha_0)\left|\widehat{P}\ominus\widehat{Q}\right| \\ &= (1-\alpha_0)\left\|\widehat{P}-\widehat{Q}\right\|_1 \;\geq\; (1-\alpha_0)(2s-2),\end{aligned}$$

with probability at least $(1-\beta_0)^3$.

Define $\Phi''P = h''(\widehat{\Phi}_\pi(hP))$, which is a multiset over $\mathbb{Z}_{O(s)}$, which we interpret as a vector in $\mathbb{Z}^{O(s)}$. By endowing this space with the $L_1$ norm, we have $\Phi'' \colon \mathbb{Z}_u \to \ell_1^{O(s)}$. By combining the above results, given point sets $P$ and $Q$, with probability at least $(1-\beta_0)^3$ we have

$$\|\Phi''P - \Phi''Q\|_1 \;\geq\; (1-\alpha_0)(2s-2) \;\geq\; \frac{3s}{2}.$$

as desired.

To prove the final assertion about $\|\Phi''P\|_1$, recall that $\widehat{P}$ is a multiset of cardinality $s$, and therefore $\Phi''P = h''\widehat{P}$ is of no greater cardinality. By interpreting

the cardinality of this set as the $L_1$ norm of a vector, we have $\|\Phi'' P\|_1 \leq s$. This completes the proof. □

Next, we consider the time needed to compute this function.

**Lemma 4.3.4** *For any $P \in \mathbb{Z}_u (\leq n)$, the function $\Phi'' P$ given in Lemma 4.3.3 can be computed in $O(n \log^3 n)$ time.*

**Proof**: The computation of $P' = h' P$ involves evaluating a simple function to each of the $n$ points of $P$, which can be done in $O(n)$ time. $P'$ is a subset of $\mathbb{Z}_s$ of cardinality at most $n$, where $s = \Theta(n \log n)$. Next, consider the computation of $\widehat{\Phi} P'$. A naive implementation based on the definition of $\widehat{\Phi}$ would involve excessive time and space. (To see this, observe that each probe involves $O(s)$ elements and must be applied to $O(s)$ distinct positions, which would yield a total time bound of $O(n^2 \log^2 n)$ to compute the results of even a single probe.)

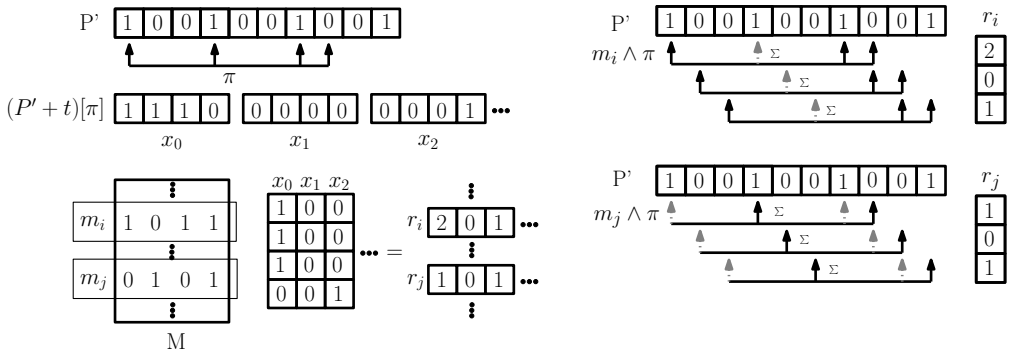


Fig. 3: Leapfrog computation with convolution operations

To achieve greater efficiency, we perform the probing and the second hash functions as a single operation. Let $\pi$ denote a probe vector, and let $\rho$ denote its

size. Recall from Lemma 4.2.6 that the second hash function maps points from $\mathbb{Z}_{2^\rho}$ to $\mathbb{Z}_{s'}$, where $s' = \Theta(s)$ and is a power of 2. This involves a random matrix $M \in \{0,1\}^{\log s' \times \rho}$ and random column vector $b \in \{0,1\}^\rho$.

Constructing $\widehat{\Phi}P'$ for each $t \in \mathbb{Z}_s$ involves computing $x_t = (P' + t)[\pi]$, and then applying the function $h''(x_t) = Mx_t + b$ (with operations performed over $\mathbb{Z}_2$). To do this, we decompose this operation into $\log s'$ operations. Let $m_i$ denote the $i$th row of $M$, and let $r_{i,t}$ be the value of the boolean inner product $(m_i \cdot x_t)$. Our objective is to compute $r_{i,t}$ for $i \in \{1, 2, \cdots, \log s'\}$. Since $m_i$ is a boolean bit-vector, we have

$$r_{i,t} \;=\; (m_i \cdot (P' + t)[\pi]) \;=\; \sum_{j=1}^{\rho} (m_i[j] \cdot ((P' + t)[\pi])\,[j])\,,$$

where $(u \cdot v)$ denotes the dot product of vectors $u$ and $v$. Observe that this a part of a convolution. Thus, given a row $m_i$, we can compute $r_{i,t}$ for all $t$ as $P' \otimes [\pi \wedge m_i]$, where "$\otimes$" denotes boolean convolution, and "$\wedge$" denotes the bitwise *and* operation.

Therefore, for all $i \in \mathbb{Z}_{\log s'}$ and $t \in \mathbb{Z}_s$ $r_{i,t}$ can be computed through $\log s'$ convolution operations. It is easy to see that $h''(x_t) = r_{*,t} + b$ (interpreted now as a binary number) where $r_{*,t} = [r_{1,t} r_{2,t} \cdots r_{\rho,t}]$. Since each convolution can be computed in time $O((s + \rho)\log(s + \rho)) = O(s \log s)$ [19], the total running time is $O(n + s \log^2 s) = O(n \log^3 n)$. This completes the proof. □

Next, we will show how to increase the success probability bounds to any desired threshold.

**Lemma 4.3.5** *Consider positive integers $n$ and $u$, where $u \leq n^c$ for some constant $c \geq 1$, a distance estimate $1 \leq \delta \leq 2n$, and failure probability $0 < \beta \leq 1$. There*

*exists* $s = \Theta(n \log n)$ *and a randomized function* $\Phi\colon \mathbb{Z}_u(\leq n) \to \ell_1^m$, *where* $m = O(n \log n \log(1/\beta))$ *that satisfies the following property. Given any two sets* $P, Q \in \mathbb{Z}_u(\leq n)$, *and letting* $\delta^*$ *denote* $\langle P \ominus Q \rangle$ *we have*

(*i*) $\|\Phi P - \Phi Q\|_1 \leq s\delta^*/\delta$, *and*

(*ii*) *if* $\delta \leq \frac{\delta^*}{8 \ln s}$ *then* $\|\Phi P - \Phi Q\|_1 \geq s$, *with probability at least* $1 - \beta$.

$\Phi P$ *can be computed in time* $O(n \log^3 n \log(1/\beta))$.

**Proof**: To increase the probability of success to $1-\beta$, we repeat the above procedure for $k = \lceil 8 \ln(1/\beta) \rceil$ trials, where each trial is performed with a different set of random choices. (All point sets being embedded use the same random choices.) We then concatenate the resulting vectors.

There is a subtlety to be noted. Thus far, we have assumed that there is a fixed value of $s$. Each invocation of Lemma 4.2.4 generates a different random prime $s$. (Recall that $s$ is $\Theta(n \log n)$, irrespective of the random choice.) To correct for the bias to the distance resulting from larger or smaller values of $s$, we take $s$ to be maximum possible value produced by the lemma, and for any smaller value $s'$ produced by invoking the lemma, we weight the associated vector by $s/s'$. As a consequence of this weighting, we may assume for the sake of simplicity, that all invocations of this lemma produce results of the same (weighted) length $s$.

Recalling that $\Phi'' P$ is the function defined in Lemma 4.3.3, we apply this $k$ times to obtain the desired vector. We define $\Phi\colon \mathbb{Z}_u(\leq n) \to \ell_1^m$, where $m = O(sk) = O(n \log n \log(1/\beta))$, to be

$$\Phi P \;=\; \frac{1}{k} \langle \Phi''_1 P, \Phi''_2 P, \ldots, \Phi''_k P \rangle .$$

Because the upper bound on $\|\Phi'' P - \Phi'' Q\|_1$ from Lemma 4.3.3 holds unconditionally (irrespective of the randomization), assertion (i) follows immediately.

To establish (ii), consider the random variables $X_i = \|\Phi''_i P - \Phi''_i Q\|_1$, for $1 \leq i \leq k$. Clearly these variables are independent and identically distributed, and $0 \leq X_i \leq 2s$. With probability at least $(1-\beta_0)^3$ we have $X_i \geq (1-\alpha_0)(2s-2)$. Therefore, we have

$$\mathrm{E}[X] \;=\; \mathrm{E}[\,\|\Phi'' P - \Phi'' Q\|_1\,] \;\geq\; (1-\beta_0)^3(1-\alpha_0)(2s-2) \;\geq\; \frac{3s}{2},$$

where the last inequality holds by our definitions of $\alpha_0$ and $\beta_0$ and of our assumption that $s \geq 128$. Let $X = \frac{1}{k}\sum_{i=1}^{k} X_i$. Since the $X_i$'s are independent, by Hoeffding's inequality [32], for any $\varepsilon > 0$ we have

$$\Pr[\,\mathrm{E}[X] - X > \varepsilon] \;\leq\; \exp\left(-\frac{2k\varepsilon^2}{(2s)^2}\right).$$

By setting $\varepsilon = s/2$ and by our choice of $k$ we have

$$\begin{aligned}\Pr[\,\|\Phi P - \Phi Q\|_1 \leq s] \;&=\; \Pr[X \leq s] \;\leq\; \Pr\left[\mathrm{E}[X] - X \geq \frac{s}{2}\right] \\ &\leq\; \exp\left(-\frac{2(8\ln(1/\beta))(s/2)^2}{(2s)^2}\right) \;=\; \beta,\end{aligned}$$

which establishes (ii).

In addition, the total computation time is $O(n \log^3 n \log(1/\beta))$, since the number of trials involving the invocation of Lemma 4.3.3 is $O(\log(1/\beta))$. □

## 4.4 Space Reduction Through Sampling

A significant shortcoming of the mapping $\Phi'' P$ given in Lemma 4.3.3 is that it generates a vector in a space of relatively large dimension, namely $O(n \log n)$. In

this section, we show how to reduce the dimension to $O(\log(1/\beta))$, where $\beta$ denotes the failure probability. Our approach will be similar to the dimension reduction technique for the Hamming distance due to Kushilevitz, Ostrovsky, and Rabani [43]. Intuitively, they show that if there is a sufficiently large difference in distance between two vectors, it is possible reduce the dimension through random sampling. For readers unfamiliar with their method here is a short summary. Let $H^d$ denote Hamming space of dimension $d$. Consider $A, B \in H^d$, and let $\ell = \|A - B\|_H$ denote their distance. (Clearly $\ell \leq d$.) For a value $r$ to be defined later, we generate a single bit from any point $A$ in $H^d$ as follows. First, we sample the coordinates of $A$ independently at random with probability $1/r$. Next, we sample from these coordinates with probability $1/2$. Finally, let $f(A)$ denote the sum these coordinates modulo 2. In the first step, with probability $(1-1/r)^\ell$ none of the coordinates upon which $A$ and $B$ disagree is sampled, implying that $f(A) = f(B)$. Otherwise, with probability $1/2$ the number of disagreeing coordinates that are sampled in the second phase is odd, and we have $f(A) \neq f(B)$. Thus, we have

$$\Pr\left[f(A) \neq f(B)\right] \;=\; \frac{1}{2}\left(1 - \left(1 - \frac{1}{r}\right)^{\ell}\right).$$

We can then convert this probability to a distance, by repeating this process some number of times and storing the results in a vector.

Although we would like to apply this approach to $\Phi'' P$, we face complications due to the fact that this is a vector of integers, not bits. By Lemma 4.3.3, the coordinates of $\Phi'' P$ are nonnegative integers whose total sum is $s$. In order to apply the above approach, we encode each coordinate of $\Phi'' P$ as an $s$-bit unary number,

which implies that we can now view it as a point set in Hamming space of dimension $s \cdot O(s) = O(s^2)$. We may then apply the above sampling function to the resulting vector.

The main lemma is stated below. We consider two cases depending on the size of $\langle P \ominus Q \rangle$, which by Lemma 4.3.3, constrains the value of $\|\Phi'' P - \Phi'' Q\|$. This constraint will allow us to apply the above method.

**Lemma 4.4.1** *Consider positive integers $n$ and $u$, where $u \leq n^c$ for some constant $c \geq 1$, a distance estimate $1 \leq \delta \leq 2n$ and failure probability $0 < \beta < 1$. There exist $s = \Theta(n \log n)$, a randomized function $\Xi \colon \mathbb{Z}_u(\leq n) \to H_1^m$, where $m = O(\log(1/\beta))$, and constants $0 < c_1 < c_2 < 1$, which satisfies the following properties. Given any two sets $P, Q \in \mathbb{Z}_u(\leq n)$, and letting $\delta^*$ denote $\langle P \ominus Q \rangle$, with probability $(1 - \beta)$:*

*(i) if $\delta \geq \delta^*$ then $\|\Xi P - \Xi Q\|_H \leq c_1 m$, and*

*(ii) if $\delta \leq \frac{\delta^*}{8 \ln s}$ then $\|\Xi P - \Xi Q\|_H \geq c_2 m$.*

*$\Phi P$ can be computed in expected time $O(n \log^3 n \log(1/\beta))$.*

**Proof**: Throughout the proof, let $\beta_0 = 1/64$. Given a fixed value $s$, we set $r = 2s$ and apply the aforementioned sampling process on $\Phi'' P$, letting $f(\Phi'' P)$ denote the result. If $\delta > \delta^*$, then by Lemma 4.3.3(i), we have $\|\Phi'' P - \Phi'' Q\|_1 \leq s$, and so by applying the sampling function to these vectors (after expanding to Hamming space) we obtain

$$\Pr\left[f(\Phi'' P) \neq f(\Phi'' Q) \,:\, \|\Phi'' P - \Phi'' Q\|_1 \leq s\right] \;\leq\; \frac{1}{2}\left(1 - \left(1 - \frac{1}{2s}\right)^s\right).$$

Let $\tau_1$ denote this probability.

Also, if $\delta \leq \frac{\delta^*}{8 \ln s}$, then by Lemma 4.3.3(ii), we have $\|\Phi'' P - \Phi'' Q\|_1 \geq \frac{3s}{2}$, with probability $(1-\beta_0)^3$. Therefore, it follows that

$$\Pr\left[f(\Phi'' P) \neq f(\Phi'' Q) \ : \ \|\Phi'' P - \Phi'' Q\|_1 \geq \frac{3s}{2}\right] \ \geq \ \frac{1}{2}\left(1 - \left(1 - \frac{1}{2s}\right)^{3s/2}\right).$$

Let $\tau_2$ denote this probability.

The remainder of the proof is similar to that of Lemma 4.3.3. We apply this $k$ times (with different random choices) to obtain the desired vector. We define $\Xi \colon \mathbb{Z}_u (\leq n) \rightarrow H^m$ (where the value of $m$ will be derived below)

$$\Xi P \ = \ \langle f_1(\Phi_1'' P), f_2(\Phi_2'' P), \ldots, f_m(\Phi_m'' P) \rangle .$$

Note that each trial involves the use of a different random prime $s$. These values in turn affect the choices of $\tau_1$ and $\tau_2$. Let $s_1, \ldots, s_m$ denote these random values, where $s_i = \Theta(n \log n)$ for all $i$. Let $\tau_{i,1}$ and $\tau_{i,2}$ denote the values of $\tau_1$ and $\tau_2$, respectively, for the $i$th trial. We shall show that the effect of changing $s$ is very small, and hence the gap between any two values $\tau_{i,1}$ and $\tau_{i,2}$ will be sufficiently large in order to apply the above method.

To prove this, we compare the upper bound of $\tau_1$ and the lower bound of $\tau_2$ over all the trials. Let $\tau_1^+ = \max_i(\tau_{i,1})$ and $\tau_2^- = \min_i(\tau_{i,2})$. Observe that $\tau_{i,1}$ and $\tau_{i,2}$ are both decreasing functions of $s$. By simple substitution in the above formula, it follows that $\tau_1^+ < 0.22$, for $s \geq 64$. Also, by consideration of the limit as $s$ tends to $\infty$, substitution into the above formula yields $\tau_2^- > 0.24$. Henceforth, we assume that $s \geq 64$, and so we have we have $\tau_2^- - \tau_1^+ \ > \ 0.02$.

Let $X_i$ denote a random indicator variable for the event that, in the $i$-th trial,

$f(\Phi''P) \neq f(\Phi''Q)$. From Lemma 4.3.5 we have

$$\Pr\left[X_i = 1 \,:\, \delta > \delta^*\right] \;=\; \Pr\left[X_i = 1 \,:\, \left\|\Phi''P - \Phi''Q\right\|_1 < s\right] \;<\; \tau_1^+.$$

As observed earlier, if $\delta \leq \frac{\delta^*}{8 \ln s}$ then $\left\|\Phi''P - \Phi''Q\right\|_1 > \frac{3s}{2}$ with probability at least $(1-\beta_0)^3$. Thus, we have

$$\begin{aligned}\Pr\left[X_i = 1 \,:\, \delta \leq \frac{\delta^*}{8 \ln s}\right] &\geq (1-\beta_0)^3 \Pr\left[X_i = 1 \,:\, \left\|\Phi''P - \Phi''Q\right\|_1 > \frac{3s}{2}\right] \\ &> (1-\beta_0)^3 \tau_2^-.\end{aligned}$$

Define a constant $c$ to be $(1-\beta_0)^3\tau_2^- - \tau_1^+$. By our bounds on $\beta_0$, $\tau_2^-$ and $\tau_1^+$, it follows that $c$ is positive. Let $X = \sum_{i=1}^m X_i$. Since the $X_i$'s are clearly independent, by Hoeffding's inequality [32], for any $\varepsilon > 0$, we have

$$\Pr\left[\mathrm{E}\left[X\right] - X > m\varepsilon\right] \;\leq\; \exp\left(-2m\varepsilon^2\right).$$

By setting $\varepsilon = \frac{c}{3} > 0$ and $m = \frac{9}{(2c)^2}\ln(1/\beta) = O(\ln(1/\beta))$, we have, for $\delta > \delta^*$,

$$\begin{aligned}\Pr\left[\left\|\Xi P - \Xi Q\right\|_H > \tau_1^+ m + \frac{c}{3}m\right] &\leq \Pr\left[X - \mathrm{E}\left[X\right] > \frac{c}{3}m\right] \\ &\leq \exp\left(-2\frac{9}{2c^2}\ln(1/\beta))(c/3)^2\right) = \beta,\end{aligned}$$

and, for $\delta \leq \delta^*/(8\log s)$, we have

$$\begin{aligned}\Pr\left[\left\|\Xi P - \Xi Q\right\|_H \;<\; (1-\beta_0)^3\tau_2^- m - \frac{c}{3}m\right] &\leq \Pr\left[\mathrm{E}\left[X\right] - X > \frac{c}{3}m\right] \\ &\leq \exp\left(-2\frac{9}{2c^2}\ln(1/\beta)(c/3)^2\right) = \beta.\end{aligned}$$

Finally, setting $c_1 = \tau_1^+ + \frac{c}{3}$ and $c_2 = (1-\beta_0)^3\tau_2^- - \frac{c}{3}$ yields the desired bounds.

The running time depends on the number of samples from $\Phi''P$. The expected number of samples is $O(s)$, since the size of the sample space is $O(s^2)$, and the sampling ratio is $\frac{1}{2s}$. Thus, the expected running time is still $O(n\log^3 n \log(1/\beta))$.

In the worst case, we can sample all $O(s^2)$ points. Thus, the running time is $O(s^2 \log(1/\beta)) = O(n^2 \log^2 n \log(1/\beta))$. □

We will now present another lemma, which will be used in the next section.

**Lemma 4.4.2** *Given the same setup as in Lemma 4.4.1, let $\delta^* = \langle P \ominus Q \rangle$. For a given distance estimate $\delta$, if $\delta^* \leq \delta$, then*

$$E\left[\|\Xi P - \Xi Q\|_H\right] \leq \frac{\delta^*}{4\delta} m.$$

**Proof**: By Lemma 4.3.3 and $\delta^* \leq \delta$, we have

$$\|\Phi'' P - \Phi'' Q\|_1 \leq s\frac{\delta^*}{\delta}.$$

Then, we apply sampling function $f$ for each $\Phi''$, from which we obtain

$$\begin{aligned}
\Pr\left[\|f(\Phi'' P) - f(\Phi'' Q)\| = 1\right] &= \frac{1}{2}\left(1 - \left(1 - \frac{1}{2s}\right)^{\|\Phi'' P - \Phi'' Q\|_1}\right) \\
&\leq \frac{1}{2}\left(1 - \left(1 - \frac{\|\Phi'' P - \Phi'' Q\|_1}{2s}\right)\right) \quad \text{(by Taylor expansion)} \\
&\leq \frac{\delta^*}{4\delta}.
\end{aligned}$$

Thus,

$$E\left[\|\Xi P - \Xi Q\|_H\right] \leq \frac{\delta^*}{4\delta} m.$$

□

## 4.5 Embedding

In Sections 4.3 and 4.4 we showed that our translation-invariant feature $\Phi$ can applied to provide a probabilistic relation between the distances between two points

set under translation. In this section, we will apply these results to show how to embed a point set into a standard metric space (e.g., $L_1$ or Hamming space). We present two results. The first is an embedding into Euclidean space under the $L_1$ metric of dimension $O(n \log^2 n)$. The distortion bounds hold with high probability. The second provides an embedding into Hamming space of dimension $O(\log n)$. The distortion bounds hold only in expectation, however. The first result is presented in Section 4.5.1 and the second in Section 4.5.2.

### 4.5.1 Embedding with High Probability

In this section we present a proof of Theorem 9. First, let $\beta_0 = \beta/(2 \log n)$. By Lemma 4.2.2, we may assume that the point sets, denoted $P$ and $Q$, have already been mapped from $\mathbb{Z}_u^d (\leq n)$ to the 1-dimensional space $\mathbb{Z}_{u'}(\leq n)$, where $u' = O(u^d)$. We then apply Lemma 4.3.5 repeatedly with failure probability $\beta_0$ and distance estimates $\delta$ ranging over $\{2^0, 2^1, 2^2, \ldots, 2^k\}$, where $k = \lceil \log 2n \rceil$. (Note that some of this notation overlaps with that used in the proof of Lemma 4.3.5, but the meanings here are quite different.) Let $\delta_i = 2^i$, and let $\Phi_i P$ denote the result of applying Lemma 4.3.5 with $\delta = \delta_i$. We apply a scalar weight to each of the resulting vectors and concatenate them to produce the following vector.

$$\Psi P \;=\; \left\langle \frac{1}{s}\Phi_0 P,\ \frac{2}{s}\Phi_1 P,\ \frac{4}{s}\Phi_2 P,\ \ldots,\ \frac{2^i}{s}\Phi_i P,\ \ldots,\ \frac{2^k}{s}\Phi_k P \right\rangle .$$

Observe that $\Psi \colon \mathbb{Z}_{u'}(\leq n) \to \ell_1^m$, where the dimension of the range of $m$ is $O(kn \log n \log(1/\beta)) = O(n \log^2 n \log(1/\beta))$.

We first establish part (i) of Theorem 9. Observe that

$$\|\Psi P - \Psi Q\|_1 \;=\; \sum_{i=0}^{k} \left\| \frac{2^i}{s}\Phi_i P - \frac{2^i}{s}\Phi_i Q \right\|_1 \;=\; \sum_{i=0}^{k} \frac{2^i}{s} \|\Phi_i P - \Phi_i Q\|_1 . \qquad (4.1)$$

Let $\delta^* = \langle P \ominus Q \rangle$. By Lemma 4.3.5(i) we have

$$\|\Phi_i P - \Phi_i Q\|_1 \;\leq\; s\delta^*/\delta_i \;=\; s\delta^*/2^i.$$

Also, $\Phi_i P$ and $\Phi_i Q$ each have at most $s$ elements, and therefore $\|\Phi_i P - \Phi_i Q\|_1 \leq 2s$. Thus we have

$$\|\Psi P - \Psi Q\|_1 \;\leq\; \sum_{i=0}^{k} \frac{2^i}{s} \min\left(2s, \frac{s\delta^*}{2^i}\right) \;\leq\; \sum_{i=0}^{k} 2^i \min\left(2, \frac{\delta^*}{2^i}\right).$$

Observe that, for $i \geq \log \delta^* - 1$, we have $\delta^*/2^i \leq 2$. Letting $k' = \lfloor \log \delta^* \rfloor$, we obtain

$$\begin{aligned} \|\Psi P - \Psi Q\|_1 \;&\leq\; \sum_{i=0}^{k'-1} 2^{i+1} + \sum_{i=k'}^{k} \delta^* \;\leq\; 2^{k'+1} + \sum_{i=0}^{k} \delta^* \\ &\leq\; 2\delta^* + \delta^*(2 + \log 2n) \;\leq\; 2\delta^* \log n, \end{aligned}$$

for all sufficiently large $n$. This establishes part (i).

To establish part (ii), Let $k'' = \lfloor \log (\delta^*/(8 \ln s)) \rfloor$. We start with Eq. (4.1). Observe that for all $i \leq k''$, we have $\delta_i \leq \delta^*/(8 \ln s)$. Therefore, by applying Lemma 4.3.5(ii), with probability $(1 - \beta)$ we have

$$\|\Psi P - \Psi Q\|_1 \;=\; \sum_{i=0}^{k} \frac{2^i}{s} \|\Phi_i P - \Phi_i Q\|_1 \;\geq\; \sum_{i=0}^{k''} \frac{2^i}{s} s \;=\; 2^{k''+1} - 1 \;\geq\; \frac{\delta^*}{8 \ln s} - 1.$$

If $\delta^*/(16 \ln s) \geq 1$ then

$$\|\Psi P - \Psi Q\|_1 \;\geq\; \frac{\delta^*}{16 \ln s}. \qquad (4.2)$$

If, on the other hand, $\delta^*/(16 \ln s) < 1$, then

$$\|\Psi P - \Psi Q\|_1 \;\geq\; 2^{k''+1} - 1 \;\geq\; 1.$$

Because $\delta^*$ is less than $16 \ln s$ and $\|\Psi P - \Psi Q\|_1 \geq 1$, the distortion is at most $16 \ln s$ ($\leq 17 \log n$) for all sufficiently large $n$. This establishes part (ii).

Finally, to establish the running time, we observe that we have invoked Lemma 4.3.5 $k = O(\log n)$ times. Each invocation takes $O(n \log^3 n \ \log(1/\beta))$ time. Thus, the total time is $O(n \log^4 n \ \log(1/\beta))$. This completes the proof.

### 4.5.2 Embedding into a Space of Logarithmic Dimension

In this section we provide a proof of Theorem 10. The proof is based on the use of Lemma 4.4.1. The price that we pay for the lower dimension is that the distortion bound holds only in expectation, not with high probability.

This proof is very similar to the proof presented in the previous section. The difference is that we will use $\Xi$ rather than $\Phi''$ and the weights. First, let $\beta_0 = \beta/(2 \log n)$. We may assume that the point sets have already been mapped from $\mathbb{Z}_u^d (\leq n)$ to the 1-dimensional space $\mathbb{Z}_{u'} (\leq n)$, where $u' = O(u^d)$. We apply Lemma 4.4.1 repeatedly, with failure probability $\beta_0$ and distance estimates $\delta$ from $\left\{2^0, 2^1, 2^2, \ldots, 2^k\right\}$, where $k = \lceil \log(2n) \rceil$. Let $\delta_i = 2^i$, and let $\Phi_i''' P$ denote the result of applying Lemma 4.4.1 with $\delta = \delta_i$. We apply an integer weight to each of the resulting vectors and concatenate them to produce:

$$\Psi' P \ = \ \left\langle \Phi_0''' P, \ 2\,\Phi_1''' P, \ 4\,\Phi_2''' P, \ \ldots, \ 2^i\,\Phi_i''' P, \ \ldots, \ 2^k \Phi_k''' P \right\rangle .$$

Observe that $\Psi' \colon \mathbb{Z}_{u'} (\leq n) \rightarrow H_1^m$, where the dimension of the range of $m$ is $O(kT) = O(\log n)$.

To establish Theorem 10(i), observe that

$$\|\Psi' P - \Psi' Q\|_1 \;=\; \sum_{i=0}^{k} \left\|2^i\,\Phi_i''' P - 2^i\,\Phi_i''' Q\right\|_1 \;=\; \sum_{i=0}^{k} 2^i\;\|\Phi_i''' P - \Phi_i''' Q\|_1\,.$$

Since $\|\Phi_i''' P - \Phi_i''' Q\|_1 \le T$ and by Lemma 4.4.2 we have

$$\begin{aligned}\mathrm{E}\left[\|\Psi' P - \Psi' Q\|_1\right] &= \sum_{i=0}^{k} 2^i\,\mathrm{E}\left[\|\Phi_i''' P - \Phi_i''' Q\|_1\right] \;\le\; \sum_{i=0}^{k'} 2^i\,T + \sum_{i=k'+1}^{k} 2^i\,\frac{\delta^*}{4\cdot 2^i}T \\ &\le 2\delta^* T + (2+\log 2n)\frac{\delta^*}{4}T \;\le\; \frac{1}{2}\log n\delta^* T,\end{aligned}$$

for all sufficiently large $n$.

In order to establish Theorem 10(ii), let $k'' = \lfloor \log\left(\delta^*/(8\ln s)\right)\rfloor$. For all sufficiently large $n$ we have

$$\begin{aligned}\mathrm{E}\left[\|\Psi' P - \Psi' Q\|_1\right] &= \sum_{i=0}^{k} 2^i\,\mathrm{E}\left[\|\Phi_i''' P - \Phi_i''' Q\|_1\right] \;\ge\; \sum_{i=0}^{k''} 2^i c_2 T \\ &\ge c_2(2^{k''+1}-1)T \;\ge\; c_2\frac{\delta^*}{17\log n}T,\end{aligned}$$

where the last inequality follows by Eq. (4.2) of Section 4.5.1.

The expected distortion is $\frac{17}{2c_2}\log^2 n < 43\log^2 n$. Thus, to obtain the desired bounds, we simply divide the embedded vector by $c_2 T$.

## 4.6 Similarity Search

We have shown how to embed point sets under translation into $L_1$ space. Since each point set is mapped into a point in this space, similarity search under translation can be reduced to (approximate) nearest neighbor searching among the embedded points. We leave the exact method for performing approximate nearest neighbor searching unspecified, but any standard method may be applied [6, 7, 38, 43]. Given

a set of $N$ points in dimension $d$, let $T(N,d)$, $S(N,d)$, $Q(N,d)$ denote the preprocessing time, the space complexity, and query time for the nearest neighbor search algorithm of interest.

First, we consider how to apply our high-probability embedding result.

**Theorem 11** *Given a database with $N$ point sets drawn from $\mathbb{Z}_u^d(\leq n)$, where $u = n^{O(1)}$, similarity search under the metric symmetric difference under translation can be reduced to the nearest neighbor search problem that the dimension of each item is $O(n \log^2 n \log(1/\beta))$. Then, the preprocessing for the similarity search is $Nn \log^3 n \log(1/\beta) + T(N, O(n \log^2 n \log(1/\beta)))$ and the space complexity is $S(N, O(n \log^2 n \log(1/\beta)))$.*

The other embedding result is to have a low dimension in expectation. We could not use it directly since it doesn't guarantee to find a proper one. Instead using only one embedding space, we use small multiple embeddings. In Section 4.4, each embedding is computed based on a given fixed distance and then sampled in order to reduce the dimension. Here we can estimate the original distance by using interactive queries That will give distortion $O(\log n)$ and sublinear space complexity.

**Theorem 12** *Given a database with $N$ point sets $\in \mathbb{Z}_u^d(\leq n)$ and $u \leq n^{O(1)}$, similarity search on symmetric difference under translation can be reduced to the nearest neighbor search problem with search on a sequence of $\lceil \log(2n) + 1 \rceil$ distances that the dimension of each item is $O(\log n \log(1/\beta))$. Then, the expected preprocessing for the similarity search is $Nn \log^3 n \log(1/\beta) + T(N, O(\log n \log(1/\beta))) \cdot \lceil \log(2n) + 1 \rceil$, and the space complexity is $S(N, O(\log n \log(1/\beta))) \cdot \lceil \log(2n) + 1 \rceil$.*

**Proof**: We will introduce an alternative way of doing similarity search, which significantly improved both dimension and distortion. We observe that the previous distortion of $O(\log^2 n)$ arises from two sources. One is from the translation invariant feature mapping, and the other arises when we weighted and concatenated the $O(\log n)$ subvectors based on geometric series of distance. Instead of concatenating the vectors, we can instead apply binary search to them in order to estimate the original distance. Although this will involve additional $O(\log \log n)$ operations (since we apply binary search on a sequence of $\lceil \log (2n) + 1 \rceil$ distances), we shall see that this allows us to reduce the distortion to $O(\log n)$. Furthermore, we will show that a dimension reduction technique can be applied for this case, and thus the space complexity can be reduced from $O(n \log^2 n \ \log (1/\beta))$ to $O(\log n \ \log (1/\beta))$. That is, the dimension is essentially logarithmic in $n$. □

Up to here, we didn't specify the value of $\beta$. If we have $N$ point sets, then we have $N^2$ pairwise distances. Thus, to obtain robust results, our embedding should have at most $O(1/N^2)$ failure probability. Therefore, we simply set $\beta = O(1/N^2)$. Moreover, $n$ is the maximum number of points among point sets. Thus, if we can partition the point sets according to $n$, we can, further, reduce the space.

## 4.7 Conclusions

We have presented a randomized algorithm that embeds an $n$-element point set over the multidimensional grid $\mathbb{Z}_u^d$, where $u$ is $n^{O(1)}$, to a single point in a multidimensional space under the $L_1$ distance. We assume that distances over $\mathbb{Z}_u^d$ are measured

using the symmetric difference under translation. This embedding has the property that with some given probability, it achieves a distortion of $O(\log^2 n)$.

This algorithm can be used in the similarity search problem with single query. If multiple queries are allowed, the distortion can be reduce to $O(\log n)$ with sublinear space of $n$. Also, we showed that, in expectation, for a single embedding technique with sampling technique can be achieved sublinear space, too.

Our existing work applies to points with integer coordinates in arbitrary dimensions and is robust to missing and spurious points. The conditions under which our embedding applies are admittedly restrictive, but to our knowledge this is the first result in embeddings that are invariant under geometric transformations and robust to outliers.

# Chapter 5

# Earth Mover's Distance under Translation

## 5.1 Introduction

The concept of the Earth Mover's Distance (EMD) was first introduced by Gaspard Monge in 1781. The name EMD was coined in 1998 by Rubner, Tomasi and Guibas [52]. The concept was introduced as a means of describing the distance between two probability distributions as a function of the effort needed to convert one into the other. For example, given two distributions, one can be seen as a pile of earth spread in space, the other as a set of holes in that same space. Then, the EMD can be thought of as the minimum amount of work needed to fill the holes with earth.

An early application of EMD was to compare two gray-scale images that may differ due to dithering, blurring, or local deformations [49]. Since then, EMD has been widely used in content-based image retrieval because it is more robust than other histogram matching techniques [52, 55, 62].

The general definition of EMD (from [52]) is as follow. Let $X$ denote a point set in $\mathbb{R}^{d\times m}$. Let $w$ denote a set of weights of $X$ in $\mathbb{R}^m$. We consider a finite

distribution $\mathcal{X}$

$$\mathcal{X} = \{(x_1, w_1), (x_2, w_2), \ldots, (x_m, w_m)\}.$$

Analogously, given a point set $Y \in \mathbb{R}^{d \times n}$ and associated weights $u \in \mathbb{R}^n$, define the distribution $\mathcal{Y}$

$$\mathcal{Y} = \{(y_1, u_1), (y_2, u_2), \ldots, (y_n, u_n)\}.$$

Given these two distributions, we define a *flow* between them to be a matrix $F = (f_{ij}) \in \mathbb{R}^{m \times n}$, where $f_{ij}$ is the amount of flow from $x_i$ to $y_j$. A flow $F$ between $\mathcal{X}$ and $\mathcal{Y}$ said to be *feasible* if

$$\begin{aligned}
f_{ij} &\geq 0 \quad i = 1, \ldots, m, \quad j = 1, \ldots, n, \\
\sum_{j=1}^{n} f_{ij} &\leq w_i \quad i = 1, \ldots, m, \\
\sum_{i=1}^{m} f_{ij} &\leq u_j \quad j = 1, \ldots, n, \text{ and} \\
\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} &= \min\left(\sum_{k=1}^{m} w_k, \sum_{k=1}^{n} u_k\right).
\end{aligned}$$

Let $\mathcal{F}$ denote the set of all feasible flows between $\mathcal{X}$ and $\mathcal{Y}$. Then, we have

$$\text{EMD}\,(\mathcal{X}, \mathcal{Y}) = \min_{F=(f_{ij}) \in \mathcal{F}(\mathcal{X}, \mathcal{Y})} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij},$$

where $d_{ij}$ is the distance from $x_i$ to $y_j$. Such a flow can be represented as a linear program and solved by any one of a number of standard methods.

It is possible to extend this distribution-based definition to the notion of the EMD distance between a pair of finite point sets $P$ and $Q$ of equal cardinality.

**Definition 4** Given two finite point sets $P$ and $Q$ of equal cardinality, let $\mathcal{M}$ denote the set of bijections from $P$ to $Q$. Then, the *earth mover's distance* between $P$ and

$Q$ is defined to be

$$\text{EMD}\,(P,Q) \;=\; \min_{M\in\mathcal{M}} \sum_{(p,q)\in M} \|p-q\|_2\,.$$

$\diamondsuit$

Computing the EMD between two point sets reduces to computing a minimum weight perfect matching in a complete bipartite graph, where the weight of each edge of $P \times Q$ is the Euclidean distance between the associated points. Thus, the EMD between two point sets of size $n$ can be computed in time $O(n^3)$ by reduction to minimum weighted perfect matching in a bipartite graph with with $2n$ vertices and $n^2$ edges. For the Euclidean planar versions, there exists $(1+\varepsilon)$ algorithm computed in $O((n/\varepsilon^3)log^6 n)$ [59].

It is straightforward to extend the definition of the EMD metric to apply to matching under translation.

**Definition 5** Given two finite point sets $P$ and $Q$ of equal cardinality, let $\mathcal{M}$ denote the set of bijections from $P$ to $Q$, and let $\mathcal{T}$ denote the space of allowable translations. Then, the *earth mover's distance between $P$ and $Q$ under translation* is defined to be

$$\text{EMD}\,\langle P,Q\rangle \;=\; \min_{t\in\mathcal{T}} \text{EMD}\,(P+t,Q) \;=\; \min_{t\in\mathcal{T}}\min_{M\in\mathcal{M}} \sum_{(p,q)\in M} \|(p+t)-q\|_2\,.$$

$\diamondsuit$

Shape-based retrieval is one of the useful applications of the EMD metric under translations. Visual similarity may not be captured by a direct comparison of the shapes due to differences in orientation or position. Thus, considering some

geometric transformations (e.g., translation and/or rotation) is useful to achieve better recall and robustness.

The goal of this chapter is to design an algorithm for point pattern searching for the EMD metric under translations. To achieve this goal, we first consider the point pattern search problem for the EMD metric (without translation). As observed earlier, one solution involves computing the distance between a query point set $Q$ and every point set $P$ of the database, but this would be very slow. Instead, our approach is based on finding a low distortion embedding into a space for which we can apply nearest neighbor searching. Our embedding will be translation insensitive, which means that it will be amenable to generalizing to the translation case. We will then show how to adapt the results of the previous chapter to provide a translation-invariant embedding.

Let us consider point sets in $\mathbb{R}^d$ of cardinality $n$. Let $\Delta$ denote an upper bound on the diameters of the point sets. For simplicity, we assume that $\Delta$ has been rounded up to a power of 2. Indyk and Thaper [40] designed an algorithm for embedding EMD in $\mathbb{R}^d$ into $\ell_1$ in $\mathbb{Z}^{d'}$, where $d'$ depends on the cardinality $n$ and the diameter $\Delta$. Under the assumption that the minimum distance between any two points is at least 1, they showed that the distortion is $O(\log \Delta)$. By merging the embedding results and Locality Sensitive Hashing (LSH), they designed an algorithm for point pattern search under the EMD metric. (Later we will describe this algorithm in greater detail.) Henceforth, we refer to this as the IT algorithm.

Cohen and Guibas [18] presented a simple method for computing the EMD under translation for two point sets. The algorithm computes a lower bound on

the EMD for all translations, and chooses the minimum one. Due to the high computation time, they proposed an efficient iterative heuristic, which may fail to produce the optimum matching, since it may fall into a local minimum.

Shirdhonkar and Jacobs [55] presented a wavelet-based variant of EMD. They show that their measure is a metric, and that it is (approximately) equivalent to the EMD in the sense that the ratio of the EMD to the wavelet EMD is bounded within some constant. This wavelet EMD metric is based on the weighted wavelet coefficients and the computation requires only linear time. They showed that the wavelet EMD can be effectively applied to content-based image retrieval. Also, they tested and compared their algorithm to other approximate EMD methods.

Ling and Okada [45] presented a fast algorithm for computing the EMD between a pair of histograms. They show that their embedding is isometric if the original underlying distance is $\ell_1$, not $\ell_2$. Then, they show that it can be computed in linear time since the number of constraints is linear. However, they did not formally establish bounds on the relation to of their embedding and the standard EMD in $\ell_2$.

Neither of these algorithms provides an efficient solution to the point pattern searching problem for the EMD metric under translations. We present a new algorithm for this problem. Our approach is to first modify Indyk and Thaper's randomized embedding algorithm to a deterministic version. In contrast to their algorithm, ours is significantly less sensitive to translation. Our embedding does involve some additional space complexity, however. Second, we show how to reduce the problem to point pattern searching with translation under the symmetric

difference distance, which we can solve by reduction to the algorithm of Chapter 4.

Before presenting our main results, we need to introduce an important assumption about the input sets. Recall that Indyk and Thaper make the assumption that the minimum distance between any two points of the union $V = P \cup Q$ is at least 1. This implies that the minimum EMD distance between $P$ and $Q$ is at least $n$, where $n = |P| = |Q|$. This assumption is important for their algorithm, since it provides a minimum scale at which point positions can be uniquely resolved and hence is basic to the approach of hierarchical grids. This assumption would seem at first glance to be ridiculous, since, for example, it rules out the possibility of computing the EMD between two point sets that share even a single point in common.

A more reasonable interpretation of this assumption is to imagine that the points have been derived from some measurement process, and the actual coordinates of the points are known to some minimal precision, determined by the sensor's limitations. Let us assume that distances have been uniformly scaled so that this minimal precision is roughly one unit. We assume that the distance between any two points (even after alignment) is defined to be the maximum of 1 and the actual distance. We call this the *unit-distance assumption*.

Recall that $\Delta$ denotes the upper bound of diameters of point sets of our database. We assume that it has been rounded up to a power of 2 for simplicity. The space $\mathcal{T}$ of allowable translations is defined to be $\mathbb{Z}^d$. Here are our main results. Recall that $|P \ominus Q|$ denotes the symmetric difference distance between two point sets $P$ and $Q$.

**Theorem 13** *Given a positive real parameter $\Delta = n^{O(1)}$ and a constant dimension $d$, there exists an embedding function $\Lambda\colon \mathbb{R}^d_\Delta(=n) \to \mathbb{Z}^m$, for $m = O(n\Delta^d \log^2 \Delta)$, such that for any two sets $P, Q \in \mathbb{R}^d_\Delta(=n)$ under the unit-distance assumption*

$$\sqrt{d} \cdot \mathrm{EMD}\,(P,Q) \;\leq\; |\Lambda P \ominus \Lambda Q| \;\leq\; 6\sqrt{d}(\log \Delta)\mathrm{EMD}\,(P,Q)\,.$$

Under our assumption that the dimension $d$ is a constant, the distortion of the embedding is $\log \Delta$. By merging this theorem and Theorem 9 (of Chapter 4), we obtain the following.

**Theorem 14** *Given a positive real parameter $\Delta = n^{O(1)}$, a constant dimension $d$, and failure probability $\beta$, there exists a randomized embedding $\Gamma\colon \mathbb{R}^d_\Delta(=n) \to \ell_1^m$, for $m = O(n\Delta^d \log^2(n\Delta) \log \frac{1}{\beta})$, such that for any two sets $P, Q \in \mathbb{R}^d_\Delta(=n)$ under the unit-distance assumption, with probability at least $(1-\beta)$*

$$\frac{\sqrt{d}}{17 \log k}\mathrm{EMD}\,\langle P,Q\rangle \;\leq\; \|\Gamma P - \Gamma Q\|_1 \;\leq\; (12\sqrt{d}\log k \log \Delta)\mathrm{EMD}\,\langle P,Q\rangle\,,$$

*where $k = 2n\Delta^d$.*

Under our assumption that $d$ is a constant, the distortion is $O(\log^2 k \log \Delta) = O(\log^2 n \log \Delta + \log^3 \Delta)$. If we assume further than $\Delta$ is polynomial function of $n$ (as we did in the previous chapter) then the distortion is $O(\log^3 n)$.

In summary, given two point sets $P$ and $Q$, we compute $\Lambda P$ and $\Lambda Q$. Then, we compute $\Psi\Lambda P$ and $\Psi\Lambda Q$, where $\Psi$ is the embedding function for symmetric difference under translation into $\ell_1$ (see Theorem 9 in Chapter 4). That is, we compose the two embedding functions, thus creating a new embedding function $\Psi \circ \Lambda$. We will establish the relationship between $\|\Psi\Lambda P - \Psi\Lambda Q\|_1$ and $\mathrm{EMD}\,\langle P,Q\rangle$.

Therefore, this result shows that we can embed a point set with translation under EMD into a vector space under the Minkowski $\ell_1$ distance.

In the remainder of this chapter, we will focus principally on giving a proof of Theorem 13. First, we review the IT embedding algorithm, and explain how to modify it to produce an initial translation-insensitive embedding. Next, we establish its distortion bounds. We shall see that this initial method is not particularly space or time efficient, however. We show how to improve the space and time complexity of this simple approach by eliminating redundancy. After this, we show how to reduce the point pattern searching problem for EMD under translations to that of symmetric difference under translations. Then, by using the result of Chapter 4, we complete the proof of the Theorem 14.

## 5.2 Translation Insensitive Embedding of the EMD into $L_1$

As mentioned earlier, Indyk and Thaper designed a randomized algorithm for embedding the EMD in $\mathbb{R}^d$ into $\ell_1$ in $\mathbb{Z}^{d'}$, where $d'$ is a function of $n$ and $\Delta$. The distortion is $O(\log \Delta)$ in expection.

For the completeness, we describe Indyk and Thaper's algorithm here. Let $P$ and $Q$ denote the point sets, recall that $n$ denotes their cardinalities, and let $\Delta$ be an upper bound on their diameters. The construction is based on imposing a collection of aligned hypercube grids in $\mathbb{R}^d$ of side lengths $\frac{1}{2}, 1, 2, 4, \ldots, \Delta$. Let $G_i$ denote a grid of side length $2^i$. Further, impose the condition that the grid $G_i$ is a refinement of grid $G_{i+1}$. The grids are all translated by a common vector chosen

uniformly at random from $[0, \Delta]^d$. For each grid $G_i$, a vector $v_i(P)$ is constructed with one coordinate per cell of the grid, where each coordinate counts the number of points in the corresponding cell. In other words, each $v_i(P)$ forms a histogram of $P$. Define a mapping $v$ by setting $v(P)$ to be the vector

$$v(P) \;=\; \left[\frac{v_{-1}(P)}{2}, v_0(P), 2v_1(P), 4v_2(P), \ldots, 2^i v_i(P), \ldots\right]$$

Note that $v(P)$ lies in an $O(\Delta^d)$-dimension space, but only $O(\log \Delta \cdot |P|)$ entries in this vector are non-zero (i.e., the vector $v(P)$ is sparse).

The randomness of their algorithm comes from the above random grid translation. The problem with applying their algorithm as a basis of a translation-invariant embedding is that, due to the hierarchical nature of the grids, point sets whose alignments differ by a quantity that is not a large power of two will likely produce dramatically different vector histograms at higher levels of the hierarchy.

To remove the sensitivity, we consider all possible shifts of the grids at all levels of the hierarchy, and then merge these results. Let $\mathbb{Z}_\Delta^d = [0, \ldots, \Delta - 1]^d$, that is, a $d$-dimensional vector space of integers modulo $\Delta$. For each translation $t \in \mathbb{Z}_\Delta^d$, we build an instance. Let $v_{[t]}(P)$ denote a vector $v(P)$ (from the IT algorithm) arising from a $t$-translated grid. (Note that we translate each grid, not the point set.) The collection of the vectors for all grid translations is denoted by $\Lambda P = \bigcup_{t \in \mathbb{Z}_\Delta^d} v_{[t]}(P)$. The trivial space complexity increases by a factor of $O(\Delta^d)$. Later, we will show how to reduce this addition space complexity factor to only $O(\log \Delta)$.

Next, we establish bounds on $\|\Lambda P - \Lambda Q\|_1$.

**Lemma 5.2.1** *Given $P, Q$ in $\mathbb{R}^d$,*

$$\sqrt{d}\Delta^d \text{ EMD}\,(P,Q) \;\leq\; \|\Lambda P - \Lambda Q\|_1 \;\leq\; 6\sqrt{d}\Delta^d \log\Delta \text{ EMD}\,(P,Q)\,.$$

**Proof**: Let us consider a pair of points $(p,q) \in (P,Q)$. Let $\delta_{pq} = \|p-q\|_2$ at level 0. Now, consider a grid at level $i$ of side length $2^i$.

If $\delta_{pq} \geq 2^i\sqrt{d}$, then it is obvious that there is no grid cell containing both $p$ and $q$. Thus, the contribution to the symmetric difference of $\Lambda P$ and $\Lambda Q$ at the level $i$ due to $p$ and $q$ is twice the total number of translated grids, that is, $2\Delta^d$.

For the other case, $\delta_{pq} < 2^i\sqrt{d}$, some grid cells may contain both $p$ and $q$, and hence, the symmetric difference may decrease. We can easily observe that only $(2^i)^d$ distinct instances exist among $\Delta^d$ instances at level $i$. Thus, each distinct instance occurs with multiplicity $\Delta^d/(2^i)^d$. Therefore, it suffices to consider all possible translations in $\mathbb{Z}^d_{2^i}$

Before counting the number of instances, let us first express the distance $\delta_{pq}$ in terms of its individual coordinates as $(\delta_1, \delta_2, \ldots, \delta_d)$. The $\ell_1$ distance is $\sum_{k=1}^d \delta_k$ and $\ell_2$ distance is $\delta_{pq} = \sqrt{\sum_{k=1}^d \delta_k^2}$. From the basic algebra, the following inequality about $\ell_1$ and $\ell_2$ is satisfied,

$$\delta_{pq} \;\leq\; \sum_{k=1}^{d} \delta_k \;\leq\; \sqrt{d}\,\delta_{pq}.$$

In order to compute the symmetric difference distance, we need to know for how many instances do both $p$ and $q$ lie within the same grid cell. Consider a $d$-dimensional hypercube with side length $2^i$. The area of the region containing both $p$ and $q$ is $(2^i - \delta_1)(2^i - \delta_2)\cdots(2^i - \delta_d)$.

Thus, the number of distinct instances which contain only $p$ is

$$2^{id} - (2^i - \delta_1)(2^i - \delta_2)\cdots(2^i - \delta_d) \;\leq\; \sum_{k=1}^{d} (2^i)^{d-1}\delta_k \;\leq\; (2^i)^{d-1} \cdot \sqrt{d}\,\delta_{pq}$$

Let $i_2$ be the minimum integer satisfying the condition $2^{i_2} \geq \delta_{pq}$. Thus, $2^{i_2} < 2\delta_{pq}$.

For all levels, the contribution of two point $p$ and $q$ to the $\ell_1$ distance is at most

$$\begin{aligned}
\|\Lambda p - \Lambda q\|_1 &\leq \sum_{i=0}^{i_2-1} 2\Delta^d \cdot 2^i + \sum_{i=i_2}^{\log\Delta} 2\frac{\Delta^d}{(2^i)^d} \cdot (2^i)^{d-1}\sqrt{d}\,\delta_{pq} \cdot 2^i \\
&\leq 2\Delta^d 2^{i_2} + \sum_{i=i_2}^{\log\Delta} 2\sqrt{d}\,\delta_{pq} \cdot \Delta^d \\
&\leq \Delta^d(4\delta_{pq} + 2\sqrt{d}\,\delta_{pq}\log\Delta) \\
&\leq \left(6\sqrt{d}\Delta^d \log\Delta\right)\delta_{pq}.
\end{aligned}$$

Summing over all pairs $(p, q)$ in the matching $M$ that define the EMD, we have

$$\begin{aligned}
\|\Lambda P - \Lambda Q\|_1 &= \sum_{(p,q)\in M} \|\Lambda p - \Lambda q\|_1 \\
&\leq \sum_{(p,q)\in M} \left(6\sqrt{d}\Delta^d \log\Delta\right)\delta_{pq} \\
&\leq 6\sqrt{d}\Delta^d \log\Delta \sum_{(p,q)\in M} \delta_{pq} \\
&\leq \left(6\sqrt{d}\Delta^d \log\Delta\right)\ \mathrm{EMD}\,(P, Q)\,.
\end{aligned}$$

Next, we establish a lower bound on $\|\Lambda P - \Lambda Q\|_1$. Recall that $v_{[t]}(P)$ denotes a vector $v(P)$ arising from a $t$ translated grid. From Indyk and Thaper's result, for any $t \in [0, \ldots, \Delta]^d$ we have

$$\left\|v_{[t]}(P) - v_{[t]}(Q)\right\|_1 \;\geq\; \sqrt{d}\ \mathrm{EMD}\,(P, Q)\,.$$

Because our construction involves $\Delta^d$ translated copies of the Indyk and Thaper construction, we have $\|\Lambda P - \Lambda Q\|_1 \geq \sqrt{d}\Delta^d \, \mathrm{EMD}\,(P, Q)$. This completes the proof.

□

Thus, we have established a translation-insensitive embedding for EMD into $\ell_1$ with distortion $6 \log \Delta = O(\log \Delta)$. Observe that the dimension of our construction is larger than that of Indyk and Thaper by a factor of $\Delta^d$. In the next section, we will consider how to reduce the dimension.

### 5.2.1 Improvement of Space Complexity and Preprocess Time

In the proof of Lemma. 5.2.1, although our construction involves $\Delta^d$ copies of the IT construction, we noticed that the number of distinct instances for the level $i$ is $\frac{\Delta^d}{(2^i)^d}$. Using this information, we change the representation of $\Lambda P$ as follows. Since we only have one distinct instance at the level 0, we multiply by $\Delta^d$ for this one distinct instance. This will not change the symmetric difference between $\Lambda P$ and $\Lambda Q$. For level 1, we have $2^d$ distinct instances, and so we multiply each by a scale factor of $\Delta^d/2^d$. In general, for level $i$, $(2^i)^d$ distinct instances exist, and so we multiply each by a scale factor of $\Delta^d/(2^i)^d$. Observe that the number of distinct instances at level $i$ is $(2^i)^d$ and each instance contributes $\Delta^d/(2^i)^d$ elements. Thus, the total number of elements at level $i$ is $\Delta^d$. Since there are $O(\log \Delta)$ levels, the total number of elements of $\Lambda P$ is $O(\Delta^d \log \Delta)$.

Let us consider an efficient way to compute each element of level $i$. Let $A_i[j_1, \ldots, j_d]$ denote the element at level $i$ corresponding to translation $(j_1, \ldots, j_d) \in$

$\mathbb{Z}^d_\Delta$, and the value of $A_i[j_1, \ldots, j_d]$ is the sum of elements of level 0 covered by the grid cell with side length $2^i$ whose origin is $(j_1, \ldots, j_d)$. That is,

$$A_i[j_1, \ldots, j_d] \;=\; \sum_{k_1=0}^{2^i-1} \cdots \sum_{k_d=0}^{2^i-1} A_0[j_1 + k_1, \ldots, j_d + k_d].$$

By the hierarchical nature of the grid, this can be computed by summing two components at the prior level. We observe that

$$A_i[j_1, \ldots, j_d] \;=\; \sum_{k_1=0}^{1} \cdots \sum_{k_d=0}^{1} A_{i-1}[j_1 + k_1 2^{i-1}, \ldots, j_d + k_d 2^{i-1}]. \tag{5.1}$$

The time complexity is $O(2^d \Delta^d \log \Delta)$, because each can computed with $2^d$ elements of the prior level.

Here is the summary of our algorithm. Initially, for all $j = (j_1, \ldots, j_d) \in \mathbb{Z}^d_\Delta$, set $A_0[j] = 1$ if there is a point at position $j$, and 0 otherwise. For $i$ from 1 to $\log \Delta$ and $j \in \mathbb{Z}^d_\Delta$, compute $A_i[j_1, \ldots, j_d]$ by Eq. (5.1). In this manner, all elements of $A$ are computed in time $O(2^d \Delta^d \log \Delta)$.

## 5.3 Similarity Search for EMD under Translations

Let us consider an algorithm for embedding a point set for EMD under translation into a vector space in $\ell_1$ metric. In the previous section, we showed how to embed a point set from EMD into the $\ell_1$ metric, without consideration of translation. Given two point sets $P$ and $Q$, $\|\Lambda P - \Lambda Q\|_1$ is can be related to EMD $(P, Q)$. We observe that all the elements of $\Lambda P$ and $\Lambda Q$ are nonnegative integers, since each component is a count that is multiplied by an integer weight. To embed such a vector into a space to which we can then apply the symmetric difference distance, we need to map

the resulting vector of integers into a $\{0,1\}$-vector (which can then be interpreted as the bit vector for a set).

Let us change the structure of $\Lambda P$ slightly by adding two additional dimensions. First, each count is transformed into unary notation, thus creating one additional dimension. For example, a vector entry whose value is 3 would be mapped to the unary vector $(1,1,1,0,0,\ldots)$. Since each coordinate of the vector $\Lambda P$ is a non-negative integer of value at most $n$, the number of coordinates in this new dimension is $n$. Next, each level can be considered as another dimension. In this way $\Lambda P$ can be viewed as a single array, rather than a collection of arrays of levels. From now on, we treat $\Lambda P$ as a binary array of dimension $d+2$. Distances are measured using the symmetric difference (or equivalently the Hamming distance over the resulting bit vectors).

In summary, given two point sets $P$ and $Q$, we compute $\Lambda P$ and $\Lambda Q$. We then compute $\Psi\Lambda P$ and $\Psi\Lambda Q$ where $\Psi$ is the embedding function for symmetric difference under translation into $\ell_1$. We will establish a relationship between $\|\Psi\Lambda P - \Psi\Lambda Q\|_1$ and EMD $\langle P, Q\rangle$. This shows that we can simply reduce the similarity search for EMD under translation to $\ell_1$.

Observe that we cannot directly apply the embedding results since $\Lambda P$ has additional two dimensions compared with $P$. Applying a translation to the original point set affects only the first $d$ dimensions of the resulting vector. After embedding, it possible, at least in principal, to apply translation to any of the dimensions. We will show that attempting to apply a translation to the additional two dimensions has no affect the distance, since any such translation will only increase distances.

In particular, given a translation $\vec{t}$ of the first $d$ components and two additional translations, $t'$ for the unary component, and $t''$ for the level component, we will show that

$$\min_{(\vec{t},t',t'')} \left\| \Lambda P_{\vec{t},t',t''} - \Lambda Q \right\|_1 \;=\; \min_{\vec{t}\in T} \left\| \Lambda P_{\vec{t},0,0} - \Lambda Q \right\|_1 ,$$

where $\vec{t} = (t_1, t_2, \cdots, t_d)$ and $P_{\vec{t},t',t''}$ denote the point set of $P$ translated by $(\vec{t}, t', t'')$.

**Lemma 5.3.1** *When $t' = 0$, the distance between two point sets achieves its minimum value.*

**Proof**: Based on the construction, a unary dimension of $\Lambda P$ and $\Lambda Q$ has at most one consecutive one from the base index (i.e., 0). We observe that the maximum overlap between two consecutive strings of ones occurs when the two are aligned at the lowest order position. This occurs when $t' = 0$. □

**Lemma 5.3.2** *When $t'' = 0$, the distance between two point sets achieves its minimum value.*

**Proof**: Recall that the sum of counts at level $i$ after weighting is $n2^{id}$, and the number of points at the leaf level are same in $P$ and $Q$. Thus, if the $t''$ coordinate of the translation vector is nonzero, we are attempting to match two different levels, and hence the distance is at least

$$n + n2^{d \log \Delta} + \sum_{i=1}^{\log \Delta - 1} n \left(2^{(i+1)d} - 2^{id}\right).$$

However, the maximum possible distance for $\|\Lambda P - \Lambda Q\|_1$ for $t'' = 0$ is $\sum_{i=0}^{\log \Delta - 1} n2^{(i+1)d}$, since the distance between $\Lambda P$ and $\Lambda Q$ is zero at the topmost level. By simple calculations, $t''$ must be zero in order to achieve the minimum distance. □

Therefore, we can apply the same translations of $P$ to $\Lambda P$ without any loss of generality. Next, we show that our embedding function is translation invariant.

**Lemma 5.3.3** *For any point set $P$ and translation $t \in \mathcal{T}$, $\Lambda(P+t) = (\Lambda P) + t$.*

**Proof**: Let $A$ denote our embedding result of $\Lambda P$. Let $x$ and $y$ denote the values of the elements $A_i(j)$ for position $j = (j_1, \ldots, j_d)$ at level $i$ in $\Lambda(P+t)$ and $(\Lambda P)+t$, respectively. By the definition,

$$A_i[j_1, \ldots, j_d] = \sum_{k_1=0}^{2^i-1} \cdots \sum_{k_d=0}^{2^i-1} A_0[j_1 + k_1, \ldots, j_d + k_d].$$

We observe that $x = y = A_i[j - t]$. Thus, $\Lambda(P+t) = (\Lambda P) + t$, as desired. □

We are now able to present the proof of Theorem 14.

**Proof**: (of Theorem 14)

Recall that $\Lambda$ is an embedding function from the EMD metric to the symmetric difference, and $\Psi$ is an embedding function from the symmetric difference under translations into $\ell_1$. By definition, we have $\mathrm{EMD}\langle P, Q\rangle = \min_{t\in\mathcal{T}} \mathrm{EMD}(P+t, Q)$. Let $t \in \mathcal{T}$ be any translation. Then, by applying Lemma. 5.2.1 to $P+t$ and $Q$, we have

$$c_1 \cdot \mathrm{EMD}(P+t, Q) \;\leq\; |\Lambda(P+t) \ominus \Lambda Q| \;\leq\; c_2 \cdot \mathrm{EMD}(P+t, Q)$$

where $c_1 = \sqrt{d}\Delta^d$ and $c_2 = 6\sqrt{d}\Delta^d \log \Delta$. By Lemma. 5.3.3, we have

$$c_1 \cdot \mathrm{EMD}(P+t, Q) \;\leq\; |(\Lambda P + t) \ominus \Lambda Q| \;\leq\; c_2 \cdot \mathrm{EMD}(P+t, Q).$$

Let $t^*$ denote the optimal translation for $P$ and $Q$ under the EMD metric, and let $\widehat{t}$ denote the optimal translation for $\Lambda P$ and $\Lambda Q$ under the symmetric difference

distance. That is,

$$t^* \;=\; \operatorname*{argmin}_{t\in\mathcal{T}} \mathrm{EMD}\left(P+t, Q\right) \qquad \text{and} \qquad \widehat{t} \;=\; \operatorname*{argmin}_{t\in\mathcal{T}} \left|(\Lambda P + t) \ominus \Lambda Q\right|.$$

Observe that

$$c_1 \cdot \mathrm{EMD}\left(P+t^*, Q\right) \;\leq\; c_1 \cdot \mathrm{EMD}\left(P+\widehat{t}, Q\right) \;\leq\; \left|(\Lambda P + \widehat{t}) \ominus \Lambda Q\right|,$$

and

$$\left|(\Lambda P + \widehat{t}) \ominus \Lambda Q\right| \;\leq\; \left|(\Lambda P + t^*) \ominus \Lambda Q\right| \;\leq\; c_2 \cdot \mathrm{EMD}\left(P+t^*, Q\right).$$

Thus, we have

$$c_1 \cdot \mathrm{EMD}\left\langle P, Q\right\rangle \;\leq\; \left\langle \Lambda P \ominus \Lambda Q\right\rangle \;\leq\; c_2 \cdot \mathrm{EMD}\left\langle P, Q\right\rangle. \tag{5.2}$$

By Theorem. 9, with probability $(1-\beta)$, we have

$$c_3 \left\langle \Lambda P \ominus \Lambda Q\right\rangle \;\leq\; \left\|\Psi\Lambda P - \Psi\Lambda Q\right\|_1 \;\leq\; c_4 \left\langle \Lambda P \ominus \Lambda Q\right\rangle, \tag{5.3}$$

where $c_3 = \frac{1}{17\log n}$ and $c_4 = 2\log n$.

Merging Eqs. 5.2 and 5.3, we obtain the desired bounds

$$c_1 c_3 \cdot \mathrm{EMD}\left\langle P, Q\right\rangle \;\leq\; \left\|\Psi\Lambda P - \Psi\Lambda Q\right\|_1 \;\leq\; c_2 c_4 \cdot \mathrm{EMD}\left\langle P, Q\right\rangle,$$

with probability at least $(1-\beta)$. To complete the proof, we define $\Gamma = \Psi \circ \Lambda$. Observe that $c_1 c_3 = \frac{\sqrt{d}}{17\log k}$ and $c_2 c_4 = (12\sqrt{d}\log k \log \Delta)$, where $k = 2n\Delta^d$. □

# Chapter 6

# Conclusions

Geometric point pattern matching problem is a fundamental computational problem and has numerous applications in various areas. In this dissertation, approximation algorithms for point pattern matching and searching have been considered. In this dissertation, three principal results were presented.

First, we considered point pattern matching between two point sets $P$ and $Q$ in the Euclidean plane under rigid transformations (translation and rotation), where dissimilarity is measured under the directed Hausdorff distance. We introduced an algorithm, called the symmetric-alignment algorithm, which operates by first computing a diametrical pair for $P$ and then computes, for every pair of distinct points of $Q$, a rigid transformation that aligns these pairs. It returns the transformation achieving the minimum Hausdorff distance. In our analysis, we introduced a geometric parameter $\rho$, called the *distance ratio*, which is defined to be half the ratio of the diameter of $P$ to the optimum Hausdorff distance between $P$ and $Q$. We showed that the approximation ratios of these algorithms can be bounded, from both above and below, as a function to the $\rho$. We proved that the approximation bounds that we achieve are nearly tight.

Second, we considered the design of a new approach for point pattern similarity search. Given a database of points sets, the objective is the preprocess the these point sets so that similarity searches involving a query point set can be performed efficiently. Our general approach is based on computing a low-distortion, transformation-invariant embedding of each point set into a metric space. By combining this embedding with any efficient nearest neighbor algorithm for the metric space, it is possible perform similarity searches efficiently. We considered this problem in the context of pattern similarity search with translation under the symmetric difference distance. In particular, we showed the existence of a translation-invariant embedding for point sets in integer space to the $L_1$ metric in real space. Through the use of projection and hashing, we first showed that this problem can be reduced to one involving 1-dimensional point sets. We then showed that, by applying random probes, a set of translation-invariant bit patterns can be generated. Based on the number of probes, the distance between the original point sets can be estimated based on the similarity of the resulting sets of bit patterns. We prove that, by applying this construction with an exponentially increasing sequence of distance estimates and appropriate weightings of the components, it is possible to construct a randomized embedding with distortion $O(\log^2 n)$. Moreover, the preprocessing time is dramatically improved by leapfrogging over intermediate steps through the use of the convolutions and the fast Fourier transform.

Finally, we considered point pattern similarity search under the earth mover's distance (EMD). We showed that, under the unit-distance assumption, the point pattern similarity search problem for point sets in real space under the EMD metric

can be embedded into integer space under the symmetric difference metric. Our embedding is translation insensitive, and achieves a distortion of $O(\log \Delta)$, where $\Delta$ is the diameter of the point set. By combining this with the above embedding for symmetric difference with translation, we obtain an embedding for the EMD metric under translation to real space under the $L_1$ metric that achieves a distortion of $O(\log^2 n \log \Delta)$.

## 6.1 Open Problems and Future Research

The work presented in this dissertation has demonstrated that point pattern similarity searching can be solved, at least approximately, through the use of embeddings into metric spaces. Our algorithms are not just heuristics, since they offer provably good approximation bounds on the quality of the results. There are, however, a number of interesting topics for future work.

### 6.1.1 Improving Performance for Point Pattern Searching

It is worthwhile considering improvements to the distortion bounds of Chapter 4. As mentioned earlier, our embedding algorithm is randomized and achieves an expected distortion of $O(\log^2 n)$ in the $L_1$ metric over a space of dimension that is roughly $O(n \log^2 n)$. We also improved the dimension to roughly $O(\log n)$, but the result holds only in expectation. A natural question is whether this distortion bound is the best achi

The most closely related work to ours is that of Cormode and Muthukrish-

nan [20], on embedding strings under edit distance with moves. They present a deterministic embedding algorithm which achieves a distortion of $O(\log n \log^* n)$ into $L_1$ with an exponential number of dimensions. Although their algorithm is not directly applicable to our problem, this suggests that lower distortion bounds may yet be achievable.

Space and time complexities are other factors. It would be interesting to know whether it is possible to achieve both logarithmic dimension with guaranteed probability bounds. Improving the preprocessing times is also an issue worth considering.

### 6.1.2 Application for Database Search

As mentioned, one of the motivating applications of our embedding algorithms of Chapters 4 and 5 is to produce an algorithm for database similarity search on point sets. The dimension of $O(n \log^2 n)$ is quite large, implying the need for relatively sophisticate data structures for nearest neighbor searching [6, 43, 48]. Thus, in order to understand the ultimate performance of our approach, it would be worthwhile to consider the simultaneous effects of the choice of embedding technique, dimension of the metric space, and the nearest neighbor search algorithm. Tradeoffs among these choices would be useful to explore.

### 6.1.3 Allowing for Noise and Other Transformations

The restriction of our embedding algorithm for symmetric difference distance can apply only to points with integer coordinates. In practice, point coordinates are the

result of measurements, and will be subject to the presence of noise and digitization errors. Handling noise is one important extension, which would be worthwhile to consider.

In the EMD embedding algorithm of Chapter 5, the algorithm works for point sets real space. However, it requires the strong unit-distance assumption, which states that the minimum distance between any two points is 1. It would be interesting to know whether this assumption can be weakened, or overcome entirely.

In addition, the embedding algorithms given in both Chapters 4 and 5 assume that the group of transformations is limited to translation. It would be very useful to extend these results to a more general group of geometric transformations, including, for example, rotation and/or scaling. Because of the discrete nature of the symmetric difference metric, upon which our results rely, such generalizations (such as rotation), which map integer coordinates to non-integer values may require new insights.

# Bibliography


[1] P. K. Agarwal, M. Sharir, and S. Toledo. Applications of parametric searching in geometric optimization. In *Proc. 3rd Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 72–82, 1992.

[2] H. Alt, O. Aichholzer, and G. Rote. Matching shapes with a reference point. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 85–92, 1994.

[3] H. Alt and L. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In *Handbook of Computational Geometry*, pages 121–153. 1999.

[4] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity and symmetries of geometric objects. *Discrete Comput. Geom.*, 3:237–256, 1988.

[5] M. Alzina, W. Szpankowski, and A. Grama. 2d-pattern matching image and video compression: Theory, algorithms, and experiments. *IEEE Trans. Image Proc.*, 11:318–331, 2002.

[6] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proc. 47th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 459–468, 2006.

[7] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. In *Proc. 5th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 573–582, 1994.

[8] M. D. Atkinson. An optimal algorithm for geometrical congruence. *J. Algorithms*, 8(2):159–172, 1987.

[9] M. Boutin and G. Kemper. Which point configurations are determined by the distribution of their pairwise distances? *Int. J. Comput. Geometry Appl.*, 17(1):31–44, 2007.

[10] L. G. Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24:325–376, 1992.

[11] D. E. Cardoze and L. Schulman. Pattern matching for spatial point sets. In *Proc. 39th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 156–165, 1998.

[12] L. Carter and M. N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

[13] F-H. Cheng. Point pattern matching algorithm invariant to geometrical transformation and distortion. *Pattern Recogn. Lett.*, 17(14):1429–1435, 1996.

[14] 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.

[15] M. Cho and D. M. Mount. Embedding and similarity search for point sets under translation. In *Proc. 24th Annu. ACM Sympos. Comput. Geom.*, pages 320–327, 2008.

[16] M. Cho and D. M. Mount. Improved approximation bounds for planar point pattern matching. *Algorithmica*, 50(2):175–207, 2008.

[17] V. Choi and N. Goyal. An efficient approximation algorithm for point pattern matching under noise. In *Latin American Theoretical Informatics Symposium*, pages 298–310, 2006.

[18] S. D. Cohen and L. J. Guibas. The earth mover's distance under transformation sets. In *Proc. 4th Annu. IEEE Int'l Conf. on Computer Vision*, pages 1076–1083, 1999.

[19] J. Cooley and J. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[20] G. Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. *ACM Trans. Algorithms*, 3(1):2, 2007.

[21] T. Dakic. *On the turnpike problem*. PhD thesis, Simon Fraser University, 2000.

[22] M. de Berg, M. van Kreveld, M. H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2nd edition, 2000.

[23] A. Efrat and A. Itai. Improvements on bottleneck matching and related problems using geometry. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 301–310. ACM Press, 1996.

[24] M. Farach-Colton and P. Indyk. Approximate nearest neighbor algorithms for hausdorff metrics via embeddings. In *Proc. 40th Annu. IEEE Sympos. Found. Comput. Sci.*, page 171, 1999.

[25] P. Finn, L. E. Kavraki, J. C. Latombe, R. Motwani, C. Shelton, S. Venkatasubramanian, and A. Yao. Rapid: Randomized pharmacophore identification for drug design. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 324–333, 1997.

[26] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981.

[27] O. Goldreich and A. Wigderson. Tiny families of functions with random properties: a quality-size trade-off for hashing. *Random Struct. Algorithms*, 11(4):315–343, 1997.

[28] M. T. Goodrich, J. S. Mitchell, and M. W. Orletsky. Practical methods for approximate geometric pattern matching under rigid motion. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 103–112, 1994.

[29] M. Hagedoorn and R. C. Veltkamp. Reliable and efficient pattern matching using an affine invariant metric. Technical Report RUU-CS-97-33, Dept. of Computing Science, Utrecht University, The Netherlands, 1997.

[30] P. J. Heffernan and S. Schirra. Approximate decision algorithms for point set congruence. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 93–101, Berlin, Germany, 1992. ACM Press.

[31] P. J. Heffernan and S. Schirra. Approximate decision algorithms for point set congruence. *Comput. Geom. Theory Appl.*, 4:137–156, 1994.

[32] W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.*, 58:13–30, 1963.

[33] J. E. Hopcroft, D. P. Huttenlocher, and P. C. Wayner. *Affine invariants for model-based recognition*. MIT Press, Cambridge, 1992.

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

[35] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:850–863, 1993.

[36] D. P. Huttenlocher and W. J. Rucklidge. A multi-resolution technique for comparing images using the Hausdorff distance. In *Proc. IEEE Conf. Comput. Vision Pattern. Recogn.*, pages 705–706. IEEE, 1993.

[37] P. Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, page 10, 2001.

[38] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 604–613, New York, NY, USA, 1998. ACM.

[39] P. Indyk, R. Motwani, and S. Venkatasubramanian. Geometric matching under noise: Combinatorial bounds and algorithms. In *Proc. 10th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 457–465, 1999.

[40] P. Indyk and N. Thaper. Fast image retrieval via embeddings. In *3rd International Workshop on Statistical and Computational Theories of Vision*, 2003.

[41] P. Indyk and S. Venkatasubramanian. Approximate congruence in nearly linear time. In *Proc. 11th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 354–360, San Francisco, 2000.

[42] R. M. Karp and M. O. Rabin. Efficient randomized pattern-matching algorithms. *IBM J. Res. Dev.*, 31(2):249–260, 1987.

[43] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 614–623. ACM Press, 1998.

[44] H. Ling and D. W. Jacobs. Deformation invariant image matching. In *Proc. 10th Annu. IEEE Int'l Conf. on Computer Vision*, pages 1466–1473. IEEE Computer Society, 2005.

[45] H. Ling and K. Okada. An efficient earth movers distance algorithm for robust histogram comparison. *IEEE Transactions on PAMI*, 29(5):840–853, 2006.

[46] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983.

[47] D. M. Mount, N. S. Netanyahu, and J. Le Moigne. Efficient algorithms for robust point pattern matching. *Pattern Recogn.*, 32:17–38, 1999.

[48] R. Panigrahy. Entropy based nearest neighbor search in high dimensions. In *Proc. 17th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 1186–1195. ACM Press, 2006.

[49] S. Peleg, M. Werman, and H. Rom. A unified approach to the change of resolution: Space and gray-level. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11:739–742, 1989.

[50] J. Rosenblatt and P. Seymour. The structure of homometric sets. *SIAM J. Alg. Disc. Methods*, 3(3):343–350, 1982.

[51] B. Rosser. Explicit bounds for some functions of prime numbers. *Amer. J. Mathematics*, 63(1):211–232, January 1941.

[52] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int'l Journal of Computer Vision*, 40:99–121, 2000.

[53] W. J. Rucklidge. Locating objects using the Hausdorff distance. In *Proc. 5th Int'l Conf. on Computer Vision*, pages 457–464. IEEE, 1995.

[54] W. J. Rucklidge. *Efficient visual recognition using the Hausdorff distance*. Number 1173 in Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1996.

[55] S. Shirdhonkar and D. W. Jacobs. Approximate earth mover's distance in linear time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[56] S. S. Skiena, W. D. Smith, and P. Lemke. Reconstructing sets from interpoint distances (extended abstract). In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 332–339. ACM Press, 1990.

[57] J. Sprinzak and M. Werman. Affine point matching. *Pattern Recogn. Lett.*, 15(4):337–339, 1994.

[58] J. Ton and A. K. Jain. Registering landsat images by point matching. *IEEE Trans. Geoscience and Remote Sensing*, 27:642–651, 1989.

[59] K. R. Varadarajan and P. K. Agarwal. Approximation algorithms for bipartite and non-bipartite matching in the plane. In *Proc. of the 10th ACM-SIAM symposium on Discrete algorithms*, pages 805–814, 1999.

[60] Z. Wang, W. Dong, W. Josephson, Q. Lv, M. Charikar, and K. Li. Sizing sketches: a rank-based analysis for similarity search. In *SIGMETRICS Perform. Eval. Rev.*, volume 35, pages 157–168, 2007.

[61] H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science and and Engineering*, 4:10–21, 1997.

[62] W. Xiong, S. H. Ong, W. Lee, and K. Foong. Local radon transform and earth mover's distances for content-based image retrieval. In *MMM'08: Proceedings of the 14th international conference on Advances in multimedia modeling*, pages 436–445, 2008.