Technische Universität München

Zentrum Mathematik

Shape Matching and Mesh Segmentation

Mathematical Analysis, Algorithms and an Application in Automated Manufacturing

Felix Schmiedl

Technische Universität München

Zentrum Mathematik

Shape Matching and Mesh Segmentation

Mathematical Analysis, Algorithms and an Application in Automated Manufacturing

Felix Schmiedl

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

eingereichten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. Boris Vexler
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. Peter Gritzmann
	2.	Univ.-Prof. Dr. Martin Henk Technische Universität Berlin
	3.	Univ.-Prof. Dr. Rolf Klein Rheinische Friedrich-Wilhelms-Universität Bonn

Die Dissertation wurde am 14. Oktober 2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 21. Januar 2015 angenommen.

Abstract

Motivated by an application in automated manufacturing, the present thesis studies optimization problems arising in several areas of computer vision. We investigate the largest common point set measure in Euclidean space, the computation of the Gromov-Hausdorff distance of metric spaces, and the dense k-subgraph problem. A mathematical analysis of these problems yields computational complexity results as well as algorithms which permit an approximate solution of large instances that cannot be solved by traditional methods.

Zusammenfassung

Die vorliegende Arbeit untersucht mehrere Optimierungsprobleme die in dem Gebiet Computer Vision auftreten und durch eine Anwendung im Bereich der Automatisierten Fertigung motiviert sind. Wir untersuchen das Largest Common Point Set Maß im Euklidischen Raum, die Berechnung der Gromov-Hausdorff Distanz von metrischen Räumen und das Dense k-Subgraph Problem. Eine mathematische Analyse dieser Probleme liefert Komplexitätsresultate sowie Algorithmen zur approximativen Lösung von sehr großen Instanzen, die nicht mit klassischen Methoden gelöst werden können.

Contents

1 Introduction **1**
- 1.1 An Application in Sheet Metal Forming . . . 1
- 1.2 Related Mathematical Problems . . . 5
- 1.3 Background and Notation . . . 7
- 1.4 Acknowledgements . . . 13

2 Affine Shape Matching **15**
- 2.1 The Largest Common Point Set Problem . . . 15
- 2.2 Wide Bases . . . 17
- 2.3 An Approximation Algorithm . . . 24
- 2.4 Pareto Approximation . . . 28
- 2.5 Adaption to the Application in Mechanical Engineering . . . 30

3 Non-Rigid Shape Matching **37**
- 3.1 The Gromov-Hausdorff Distance . . . 37
- 3.2 Computational Complexity . . . 43
- 3.3 Algorithmic Approaches . . . 56
- 3.4 Partial Matching . . . 86

4 Dense k-Subgraph Problem **91**
- 4.1 An Application in Forestry . . . 93
- 4.2 Threshold-based Pruning and Approximate Solution . . . 95
- 4.3 An Optimal Threshold . . . 100
- 4.4 Empiric Results . . . 102

5 Mesh Segmentation **107**
- 5.1 Minimal Covering of a Mesh . . . 108
- 5.2 Subdivision in Submeshes . . . 110
- 5.3 Conclusion . . . 112

Index **113**

Bibliography **115**

Chapter 1

Introduction

Computers and data processing have long superseded human capabilities in many sectors and revolutionized our daily lives. In areas which deal with context sensitive information, however, there is still a lot of room for improvement. One of these areas is computer vision, a discipline in computer science which deals with visual information and other high-dimensional data.

In the present thesis, we investigate the possibility of automatizing a traditionally manual manufacturing process in engineering. To achieve this, we study variants of the *shape matching* and *mesh segmentation* problem from computer vision using mathematical tools, mainly based on combinatorial optimization.

At first, we give a short overview of the interdisciplinary project in whose periphery this thesis originated. Next, the main mathematical challenges are identified and subsequently addressed in the following chapters, where we give several original results on the complexity of these problems as well as algorithms which give at least approximate solutions.

1.1 An Application in Sheet Metal Forming

While the introduction of industrialized mass production was a major breakthrough in production technology in the early 20th century, the demand in the beginning of the 21st century is shifting gradually towards individualized products. This poses a major challenge for established production methods, which usually require individualized tool sets and a significant amount of manual labour for each product line. For small batch or individualized production, these processes cannot be used economically.

To adapt to this demand, alternative methods which are more suitable for these requirements have been investigated. In the joint research project [Vol+13] with the Institute of Metal Forming and Casting at Technische Universität München, the goal was to investigate possibilities for the automation of a traditionally manual method for the individualized forming of sheet metal parts. This method is described in Section 1.1.1

A first step towards the automation of this method, a manufacturing system called *copied driving*, was introduced in the preceding project [Hof+10]. The manual crafting process is tracked by a camera system and suitably processed for a robot controlling the process, which can subsequently reproduce the manufacturing steps performed by the worker. This data set is referred to as *manufacturing strategy*. Due to the manual production process required initially, copied driving is not suitable for individual or prototypic applications and still requires a specialized operator. In the follow-up project [Vol+13], the main goal was to overcome these limitations in a joint effort of mechanical engineering and mathematics. By deriving manufacturing strategies for new component shapes from known strategies, a new concept enabling automated production of individualized components is realized. The system utilizes a

database of already crafted components and their manufacturing strategies, which we further on refer to as *standard elements*. In addition to the exact reproduction of sheet metal parts, one of the goals in the engineering part was to enable the production of transformed standard elements for a class of feasible transformations to increase the utility of the process.

The overall approach is illustrated in the flow diagram depicted in Figure 1.1. After digitizing the desired component geometry, it is segmented into parts of suitably transformed standard elements. Since the number of standard elements is limited, it is necessary to be able to handle a wide bandwidth of possible transformation strategies for these parts in the segmentation step as well as the production process. The main mathematical challenges arise in the segmentation step and are described in Section 1.2.

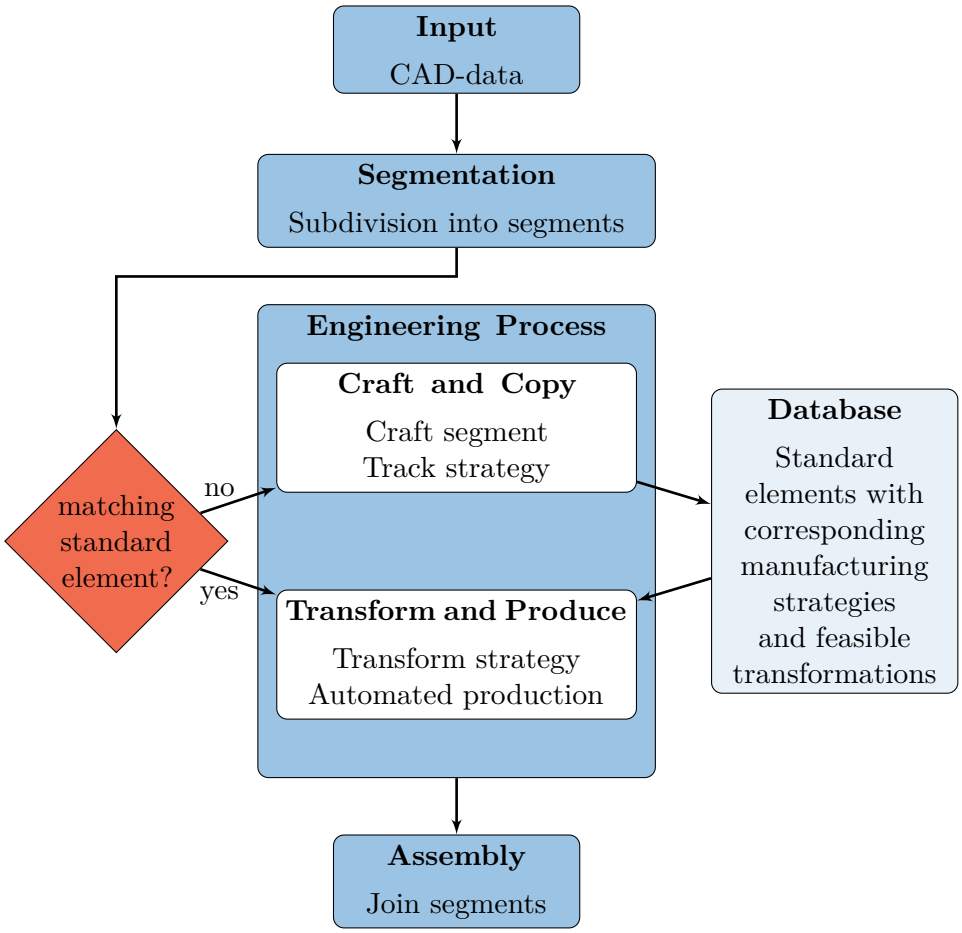


Figure 1.1: Flow chart for the automated driving concept utilizing a database of manufacturing strategies.

An appropriate manufacturing strategy can then be devised and applied for automated production. For parts which cannot be produced using the current database, the corresponding segment is crafted manually while recording the manufacturing strategy and immediately included in the database. Hence, a learning strategy is implemented which increases the power of the pool of standard geometries sustainably.

1.1.1 Forming Method

In the field of sheet metal forming, the demand for individualization is attended to mostly by incremental methods. In contrast to conventional methods, which allow for the forming of a sheet metal part in one passage, incremental methods employ a series of consecutive shaping steps to form the desired component geometry. These can be implemented with general purpose tools independent of the target geometry, while traditional methods rely on custom-built dies for each produced form as well as expensive machinery. By eliminating the need for these one-of-a-kind tool sets for each component geometry, incremental methods can be used economically for small batch sizes, where the longer production time for individual parts can be tolerated.

We consider a special kind of incremental forming process — called driving process — which enables the production of almost any desired component shape. The process is carried out on simple and inexpensive gap frame presses and solely makes use of universal tool sets for the deformation of the sheet metal. Each tool set consists of a pair of top and bottom tools, which are small compared to the sheet metal part. A single forming step consists of clamping the sheet metal part between the top and bottom tools at a certain position. From an engineering point of view, the tool sets that can perform local stretching and local shrinking in the deformed zone are of particular interest.

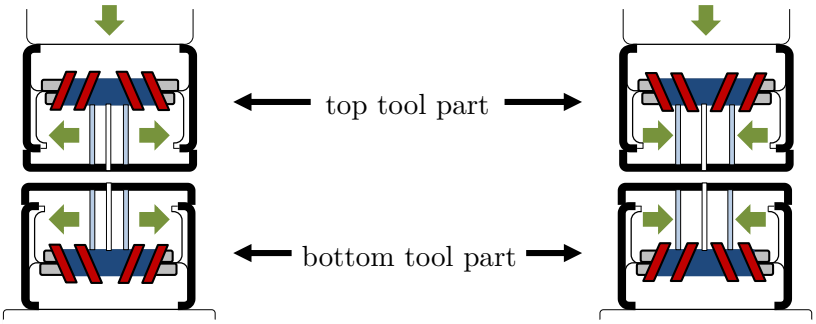


Figure 1.2: A schematic sketch[1] of the principle of local material stretching (left) and shrinking (right) with appropriate tool sets for the driving process.

The principle of these tools is schematically shown in Figure 1.2. There are mounting parts in the top and bottom tool which enable a transfer of the vertical stroke force into a horizontal motion. Hence, the split tool jaws are actuated while the sheet metal is clamped through the vertical pressure and the resulting friction. The jaws of the stretching tool move apart and thus cause local stretching of the material, whereas the shrinking tools provoke localized shrinking by movement of the jaws towards each other.

The shrinking and stretching operations do not only cause obvious local deformations of the sheet metal, but also alter intrinsic properties like tension and density of the material which results in a global deformation of the geometry. While a theoretical analysis and simulation of a single individual forming step has been considered [HHP05], the driving process in general has not yet been fully understood.

In practice, desired sheet metal parts are manufactured in an interactive process of consecutive

[1] Image courtesy of Daniel Opritescu, Institute of Metal Forming and Casting, Technische Universität München

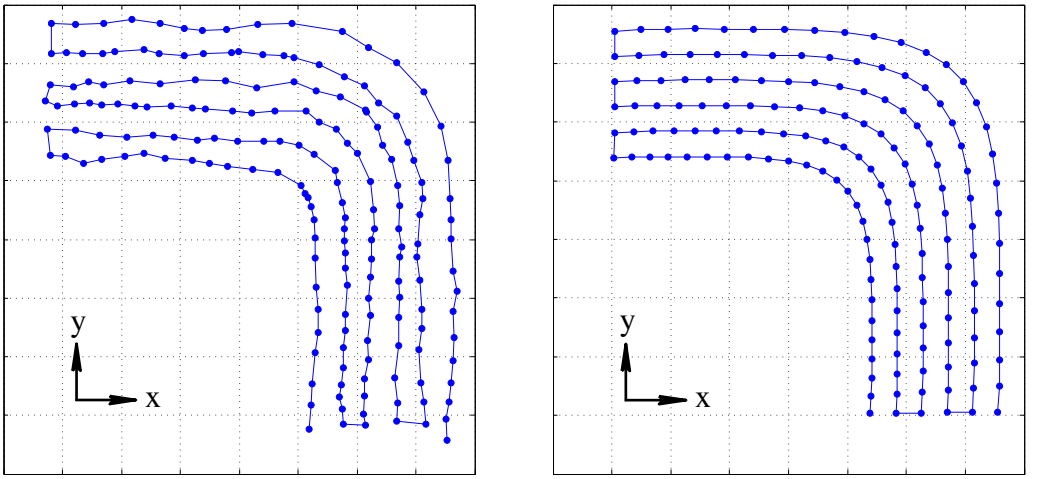
evaluation and refinement of the part in production. Both steps are generally carried out in manual labour and the process heavily relies on the skills of the operator. Since there is no auxiliary material besides gauges, the experience and eye sight of the operator are crucial for the evaluation and refinement step.

This process allows the creation of almost any geometry, but being an interactive, manual production method, it suffers from drawbacks like poor reproducibility and high manual effort. Due to the complexity of the process and the need for highly trained operators, this manufacturing process is to date used mainly in technical areas where only small batches are produced and the high amount of manual labour can be tolerated. Examples include prototyping, restoration of vintage cars, and the manufacturing of ships and (small) aircraft.

1.1.2 Transformation of Standard Elements

The presented concept of automated driving strongly depends on an efficient use of the database, i. e., it should contain few elements which suffice to produce a great variety of geometries. Hence, it is inevitable to consider standard elements in a parameterized way. The process should not only allow for the exact reproduction of parts stored in the database but should also enable geometric variations of the parts.

The copied driving system only allows for the exact reproduction of parts using the tracked manufacturing strategy in the specific dimensioning. In a first step, we analyse the employed manufacturing strategy and subsequently approximate it by an analytic parameterized description, cf. Figure 1.3. The second step consists in recreating the component geometry by utilizing a generated manufacturing strategy with an appropriate setting of parameters in order to allow for substitution of the tracked manufacturing strategy by the analytic description, see [Opr+12; OV14]. Additionally, further variations, such as scaled or sheared components, can be wrought by a transformation of the manufacturing strategy and suitable adjustment of parameters, cf. [Vol+13].



(a) Tracked manufacturing strategy

(b) Analytically generated manufacturing strategy

Figure 1.3: Manufacturing strategies for a sample component in sheet metal part coordinate system[1]; dots represent stroke positions correlating to strokes on the component with the tool set.

This approach has been validated by adopting the method on a sheet metal part serving as a sample standard geometry. Scaled variants of the sample geometry can be reproduced within system tolerances for scaling factors ranging from 0.80 to 2.00. The tolerances are strongly system dependent, as the main contributions arise from deviations in robot positioning and from digitizing and measuring the part geometry by the optical system.

A suitable adjustment of the process parameters with respect to the scaling factors enables the manufacturing system to produce user-defined scaled components. The adoption of other geometric variation types, including a subclass of common scaling and shearing, has also been implemented. A detailed description of the procedure is given in [Vol+13]. A mathematical characterization of the achievable transformations is given in Section 2.5.

1.2 Related Mathematical Problems

The mathematical challenge in this engineering problem lies in the suitable segmentation of a possibly complex part, such that each segment can be produced by a feasible transformation of one of the given standard elements. The parts are represented as 2-dimensional mesh surfaces. We present a three stage method for this problem, which takes the specific requirements of the problem setting into account.

Considering the restriction of the manufacturing system to a relatively small number of standard elements in the database, it is inevitable to incorporate information about which parts of the component can be represented by transformed standard elements in the segmentation step. Therefore, the first step consists of determining occurrences of transformations of standard elements in the component, see Figure 1.4.

This task, usually referred to as *registration* or *shape matching*, has numerous applications in various scientific and economical fields. The parts are reduced to point clouds for the registration step. This allows for faster computational methods and the loss of accuracy is negligible in practice. The task can then be formulated as follows: given two point clouds $P, Q \subseteq \mathbb{R}^d$ representing a standard element and the desired sheet metal part, the goal is to find a transformation optimally aligning P to Q with respect to a given measure, such as the Hausdorff distance. Numerous approaches for various types of transformations have been considered. One of the best-known is described in [BM92], which tries to find an aligning transformation which is Euclidean, i. e., consists of a combination of translation, rotation and reflection, of two point clouds by iterative refinement. Unfortunately, this method depends on an approximate initial alignment of the data to obtain correct results and is thus not suitable for our purposes. In this thesis, we develop a method which computes an aligning affine transformation with certain additional properties for two point clouds with bounded error. These transformations also represents the current state of the art concerning feasible transformations of manufacturing strategies of sheet metal parts, see [Vol+13]. The theory and implementation of this algorithm are detailed in Chapter 2 and partly based on joint work with Peter Gritzmann and Michael Ritter [GRS14].

The success of shape matching methods in applications has also sparked a lot of interest in the more general problem of so-called *non-rigid shape matching*. In contrast to traditional (rigid) shape matching, where one tries to find a (typically Euclidean or affine) transformation of the ambient space that minimizes the Hausdorff distance of the given point clouds, non-rigid shape matching is concerned with the similarity of shapes in a more general context. Consider the two shapes depicted in Figure 1.5. For a human, it is clear that these shapes are very

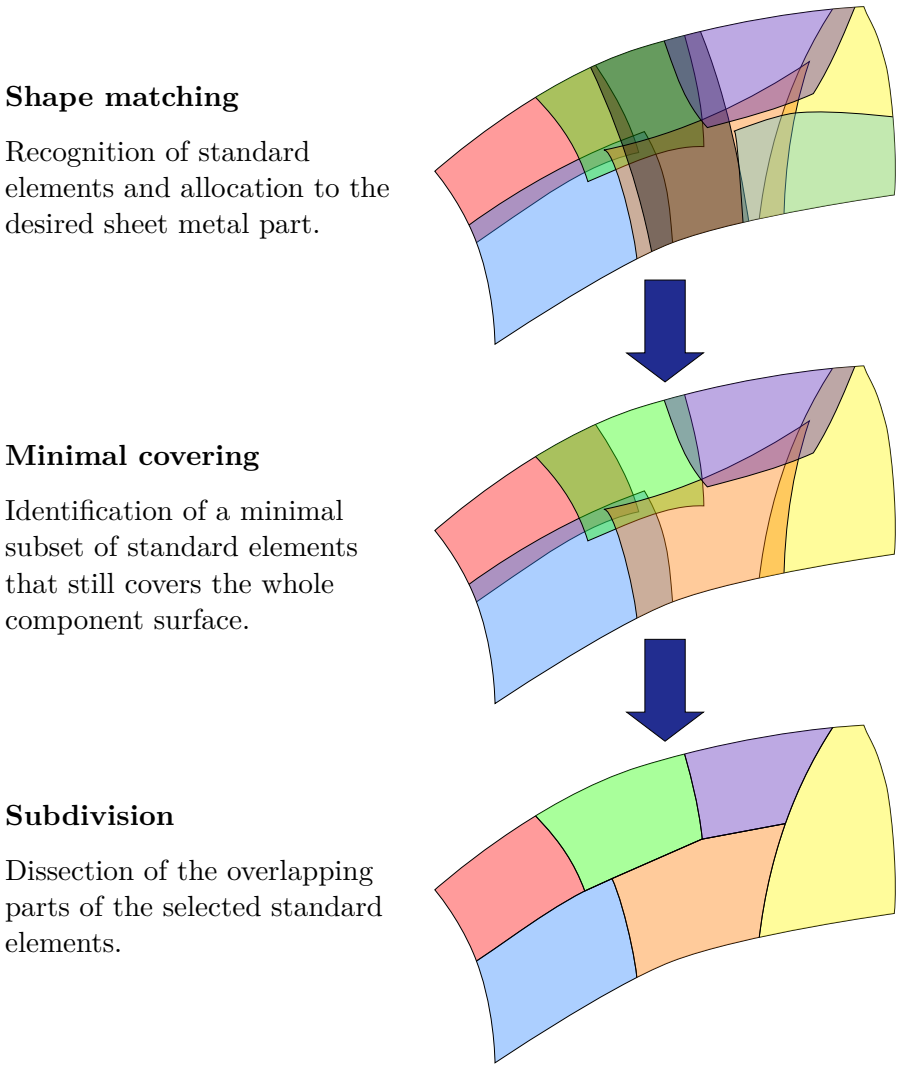


Figure 1.4: Flow chart of the segmentation process.

similar. In fact it is the same person, just in different poses. But this similarity is not reflected by the usual measures used for rigid shape matching.

An approach to this problem, which was pioneered by [EK01; Bro+04; MS05], is to consider the shapes as separate metric spaces and not as subsets of a common ambient space. This is motivated by the observation that the surface (e. g., skin, sheet metal, etc.) of the objects we intend to compare is usually not too elastic. Therefore, the geodesic distances between points on the surface do not change too much under the transformations we consider feasible. This could also be of interest for the original problem of sheet metal forming in future projects: computing a matching between two surfaces yields the amount by which parts of one surface need to be stretched or shrunk to be transformed to the other surface. These results could be used as basis for subsequent research investigating more general transformations of the shapes in the database, which would render the method even more effective. Chapter 3 investigates the computational complexity of measuring the difference between metric spaces and gives

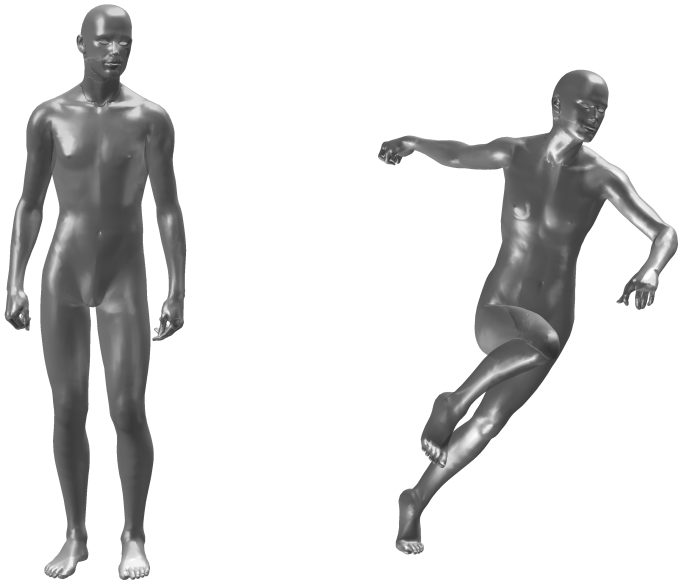


Figure 1.5: A human shape[2] in different poses.

algorithms that compute a *correspondence* with low distortion between two metric spaces. The results presented in this chapter are based on the manuscripts [Sch14b; Sch14a].

Furthermore, we observe a connection between non-rigid shape matching and a well-known problem in combinatorial optimization. In Chapter 3, we show that the non-rigid shape matching problem can be formulated as *dense k-subgraph* problem, which asks for a subgraph with maximal edge weight on exactly k vertices of a given graph. This problem is investigated in Chapter 4, which is based on joint work with Steffen Borgwardt [BS14]. We present a (non-polynomial time) $\left(1 + \frac{1}{k-1}\right)$-approximation algorithm, which exploits the fact that many nodes can be disregarded for a solution in certain graphs. Moreover, we present an additional application in forestry, for which this approach is very successful.

After surface registration has been performed, we assume that the desired geometry is covered by appropriately transformed standard elements. Otherwise, suitable parts are added to the database. As a next step, Section 5.1 details how to compute a *minimal covering*, i. e., a subset of the standard elements that still covers the whole surface. Note that the selected standard elements may still have overlapping parts. A segmentation of the component geometry and hence a suitable manufacturing strategy can then be derived by *subdivision*, which is described in Section 5.2. Overlapping parts are assigned to exactly one of the fitting standard elements, while taking the difficulty of cutting the specific geometry of the sheet metal parts into account.

1.3 Background and Notation

To describe the above mentioned problems rigorously, we first introduce some notation which is used throughout this thesis.

[2]Surface mesh distributed as part of the TOSCA project, see [BBK08].

The number of elements in a set X is denoted by $|X|$. The power set 2^X of X is defined as the set of all subsets of X, i. e., $2^X \coloneqq \{A \mid A \subseteq X\}$. The symmetric difference of two sets X, Y is denoted by $X \triangle Y \coloneqq (X \setminus Y) \cup (Y \setminus X)$. For a set X and a function $f\colon X \longrightarrow \mathbb{R}$, we denote by $\operatorname{argmin}_{x \in X} f(x)$ the set of all points which minimize f. We also use

$$x^* \coloneqq \operatorname*{argmin}_{x \in X}{}_1 \, f(x)$$

to denote an arbitrary but fixed element $x^* \in \operatorname{argmin}_{x \in X} f(x)$; argmax and argmax_1 are defined accordingly. For $A \subseteq X$, we use the notation $f|_A$ to refer to function f restricted to A.

Euclidean Geometry

Furthermore, we denote by $e_i \coloneqq (0, \ldots, 0, 1, 0, \ldots, 0)^T \in \mathbb{R}^d$ the ith unit vector of $\mathbb{R}^d$ and use $\mathbb{1}_d$ to refer to the d-dimensional vector $(1, \ldots, 1)^T \in \mathbb{R}^d$. The d-dimensional identity matrix is denoted by $I_d \in \mathbb{R}^{d \times d}$ and the scalar product of $\mathbb{R}^d$ is denoted by $\langle \cdot, \cdot \rangle$, i. e., the scalar product of $v, w \in \mathbb{R}^d$ is given by

$$\langle v, w \rangle \coloneqq v^T w = \sum_{i=1}^{d} v_i w_i.$$

The Euclidean norm on $\mathbb{R}^d$ is defined by $\|x - y\|_2 \coloneqq \langle x - y, x - y \rangle$ and $\|\cdot\|$ denotes an arbitrary norm on $\mathbb{R}^d$. For $p \in \mathbb{N} \cup \{\infty\}$, the vector space $\mathbb{R}^d$ endowed with the p-norm $\|x\|_p \coloneqq \sqrt[p]{|x_1|^p + \ldots + |x_d|^p}$, $x \in \mathbb{R}^d$, is referred to as L_p.

For a set $A \subseteq \mathbb{R}^d$, we use $A^\perp \coloneqq \left\{ x \in \mathbb{R}^d \mid \langle x, a \rangle = 0 \; \forall a \in A \right\}$ to denote the set of normals of A. The affine hull of A is given by

$$\operatorname{aff}(A) \coloneqq \left\{ \sum_{a \in A} \lambda_a a \mid \lambda_a \in \mathbb{R} \; \forall a \in A, \; \sum_{a \in A} \lambda_a = 1 \right\}$$

and

$$\operatorname{conv}(A) \coloneqq \left\{ \sum_{a \in A} \lambda_a a \mid \lambda_a \in [0, 1] \; \forall a \in A, \; \sum_{a \in A} \lambda_a = 1 \right\}$$

defines the convex hull of A. The dimension $\dim(A)$ of a A denotes the dimension of the affine subspace spanned by A

$$\dim(A) \coloneqq \dim(\operatorname{aff}(A)),$$

and A is called affinely independent if $\dim(A) = |A| - 1$.

In the context of shape matching, we use the notion of Euclidean or rigid transformations, which are characterized as isometric maps from $\mathbb{R}^d$ to $\mathbb{R}^d$. These transformations are a combination of a rotation, translation and reflection.

Similarity of Shapes

In rigid shape matching, the Hausdorff distance and its directed version are some of the most commonly used measures for the similarity of shapes. Intuitively, two sets have a small Hausdorff distance if each point in either set is close to some point in the other set. To be able to measure the distance between two points, we consider metric spaces.

Definition 1.1 (Metric space)
Let X be a set and $d_X \colon X \times X \longrightarrow \mathbb{R}$. The tuple (X, d_X) is called metric space if d_X is a metric on X, i. e., if the following properties hold for all $x, y, z \in X$:

i) $d_X(x, y) \geq 0$ and $d_X(x, y) = 0 \Leftrightarrow x = y$

ii) $d_X(x, y) = d_X(y, x)$ (symmetry)

iii) $d_X(x, z) \leq d_X(x, y) + d_X(y, z)$ (triangle inequality).

We call a metric space (X, d_X) rational if X is finite and $d_X(x, y) \in \mathbb{Q}$ for all $x, y \in X$. A ball with centre $x \in X$ and radius $\varepsilon \geq 0$ is denoted by $\mathbb{B}_x(\varepsilon) \coloneqq \{x' \in X \mid d_X(x', x) \leq \varepsilon\}$.

The maximal distance of two elements of X is called the *diameter* of X:

$$\operatorname{diam}(X) \coloneqq \sup_{x,y \in X} d_X(x, y)$$

Two metric spaces are called isometric if there exists an isometry $\phi : X \longrightarrow Y$, i. e., if ϕ is bijective and

$$d_X(x_1, x_2) = d_Y(\phi(x_1), \phi(x_2))$$

holds for all $x_1, x_2 \in X$.

The original definition of the Hausdorff distance was supposedly given by Felix Hausdorff in his famous book “Grundzüge der Mengenlehre” [Hau14].

Definition 1.2 (Hausdorff distance)
Let $P, Q \subseteq X$ be subsets of a metric space (X, d_X). The directed Hausdorff distance of P and Q is given by

$$d_h(P, Q) \coloneqq \sup_{p \in P} \inf_{q \in Q} d_X(p, q).$$

The Hausdorff distance is then defined by

$$d_H(P, Q) \coloneqq \max\{d_h(P, Q), d_h(Q, P)\}.$$

If (X, d_X) is compact, d_H is a metric on the set of compact subsets of X.

In computer vision, another measure which is frequently used is the so-called bottleneck distance introduced in [EI96]. In contrast to the Hausdorff distance, it requires a one-to-one mapping from P to Q.

Definition 1.3 (Bottleneck distance)
Let $P, Q \subseteq X$ be subsets of a metric space (X, d_X) with $|P| = |Q|$. The bottleneck distance of P and Q is defined by

$$d_B(P, Q) \coloneqq \inf\left\{ \sup_{p \in P} d_X(p, \pi(p)) \mid \pi \colon P \longrightarrow Q \text{ is a bijection} \right\}.$$

In Chapters 2 and 3, we investigate other measures for the (partial) similarity of shapes. Therefore, it is necessary to describe which point in one shape corresponds to which point in the other shape. In general, it is not sufficient to model this by maps. We use relations instead.

Definition 1.4 (Relation)
Let X, Y be two sets. A subset $R \subseteq X \times Y$ is called a relation between X and Y. We refer to an element $(x, y) \in R$ as an assignment of x to y and we say that x is matched or assigned to y. Further, we denote by

$$\mathcal{P}_1(R) \coloneqq \{x \in X \mid \exists y \in Y \text{ such that } (x, y) \in R\}$$

and

$$\mathcal{P}_2(R) \coloneqq \{y \in Y \mid \exists x \in X \text{ such that } (x, y) \in R\}$$

the projection of R to X and Y, i. e., the subsets of X and Y *covered* by R, respectively.

A relation $R \subseteq X \times Y$ which matches a point $x \in X$ to at most one point in Y and a point $y \in Y$ to at most one point in X is referred to as *bottleneck* relation.

A relation which covers both X and Y is called correspondence.

Definition 1.5 (Correspondence)
Let X, Y be two sets. A relation $R \subseteq X \times Y$ is called correspondence between X and Y if

$$\mathcal{P}_1(R) = X \text{ and } \mathcal{P}_2(R) = Y,$$

i. e., if both X and Y are covered by R. The set of all correspondences is denoted by $\mathcal{R}(X, Y)$.

A correspondence $R \in \mathcal{R}(X, Y)$ which is also a bottleneck relation, i. e.,

$$\forall x \in X \; \exists_1 y \in Y \text{ such that } (x, y) \in R$$
$$\forall y \in Y \; \exists_1 x \in X \text{ such that } (x, y) \in R$$

is called bottleneck correspondence. In this case, each point $x \in X$ is assigned to exactly one point $y \in Y$ and vice versa, i. e., R induces the bijection $\pi \colon X \longrightarrow Y$, $\pi(x) = y \Leftrightarrow (x, y) \in R$. The set of bottleneck correspondences is denoted by $\Pi(X, Y)$.

The Hausdorff distance and bottleneck distance can be easily formulated using this notation:

Lemma 1.6
Let $P, Q \subseteq X$ be compact subsets of a compact metric space (X, d_X), then

$$d_H(P, Q) = \inf_{R \in \mathcal{R}(P,Q)} \sup_{(p,q) \in R} d_X(p, q)$$

and

$$d_B(P, Q) = \inf_{R \in \Pi(P,Q)} \sup_{(p,q) \in R} d_X(p, q).$$

Proof. For compact $P, Q \subseteq X$, a correspondence inducing the Hausdorff distance is given by

$$R \coloneqq \left\{(p, q) \mid p \in P,\ q = \operatorname*{argmin}_{q' \in Q} d_X(p, q')\right\} \cup \left\{(p, q) \mid q \in Q,\ p = \operatorname*{argmin}_{p' \in P} d_X(p', q)\right\}$$

and this is best possible. The existence of $\operatorname*{argmin}_{q' \in Q} d_X(p, q')$ and $\operatorname*{argmin}_{p' \in P} d_X(p', q)$ is guaranteed since P and Q are compact.

An optimal bottleneck correspondence $R \in \Pi(X, Y)$ induces the bijective map $\pi \colon P \longrightarrow Q$, $\pi(p) = q \Leftrightarrow (p, q) \in R$, which minimizes the maximal distance between a point and its image and vice versa. □

The following lemma shows that it is sufficient to consider correspondences whose size is bounded by the sum of the sizes of the metric spaces.

Lemma 1.7
Let X, Y be finite sets and $R \in \mathcal{R}(X, Y)$ a correspondence. There exists $R' \in \mathcal{R}(X, Y)$ with $R' \subseteq R$ and $|R'| \leq |X| + |Y|$.

Proof. For each $x \in X$, there exists $y_x \in Y$ with $(x, y_x) \in R$. Similarly, for each $y \in Y$, there is $x_y \in X$ such that $(x_y, y) \in R$. Therefore,

$$R' \coloneqq \{(x, y_x) \in X \times Y \mid x \in X\} \cup \{(x_y, y) \in X \times Y \mid y \in Y\}$$

is a correspondence with $R' \subseteq R$ and $|R'| \leq |X| + |Y|$. □

The thickness of the sheet metal parts we consider is very small in relation to their surface and can therefore be neglected. The objects we deal with in the engineering applications can then be represented as 2-dimensional surfaces. For the use in algorithms, we assume that they are given as mesh surfaces, which are defined in the following.

Meshes

A mesh surface is a discrete representation of a surface used in computer graphics. In this thesis, we consider only triangle meshes, but generalizations are easily possible. A triangulation of a set in $\mathbb{R}^d$ is given by a simplicial complex, see [Mau96, p. 33].

Definition 1.8 (Simplicial complex)
Let $n \in \mathbb{N} \cup \{0\}$ and $S \subseteq \mathbb{R}^d$ be an affinely independent set with cardinality $|S| = n + 1$, i. e., $\dim(S) = n$. The convex hull $\sigma \coloneqq \operatorname{conv}(S)$ of S is called n-simplex. For $T \subseteq S$ and $m \coloneqq \dim(T)$, the m-simplex $\tau \coloneqq \operatorname{conv}(T)$ is called m-face of σ. A finite collection $\mathcal{S}$ of simplices in $\mathbb{R}^d$ is called (geometric) simplicial complex if

- $\sigma \in \mathcal{S}$ and $\tau \subseteq \sigma$ is an m-face of σ for some $m \in \mathbb{N} \cup \{0\} \implies \tau \in \mathcal{S}$
- $\sigma, \tau \in \mathcal{S} \implies \sigma \cap \tau$ is either the empty set or an m-face of both σ and τ for some $m \in \mathbb{N} \cup \{0\}$.

The union of all simplices $\mathcal{U}(\mathcal{S}) \coloneqq \bigcup_{\sigma \in \mathcal{S}} \sigma$ is called underlying space of $\mathcal{S}$.
A topological space X for which there is a simplicial complex $\mathcal{S}$ and a homeomorphism $\phi \colon \mathcal{U}(\mathcal{S}) \longrightarrow X$ from the underlying space $\mathcal{U}(\mathcal{S})$ of $\mathcal{S}$ to X is called polyhedron. The tuple $(\mathcal{S}, \phi)$ is called triangulation of X.

A mesh is a special case: if the underlying space $\mathcal{U}(\mathcal{S})$ of a simplicial complex $\mathcal{S}$ is a surface in $\mathbb{R}^3$, $\mathcal{S}$ is called triangle surface mesh or mesh, cf. [Hop96].

Definition 1.9 (Mesh)
Let $\mathcal{S}$ be a simplicial complex in $\mathbb{R}^3$. $\mathcal{S}$ is called triangle surface mesh or mesh if its underlying space $\mathcal{U}(\mathcal{S})$ is a surface in $\mathbb{R}^3$. A point $v \in \mathbb{R}^d$ is called vertex of the mesh if $\{v\}$ is a 0-simplex in $\mathcal{S}$, i. e., $\{v\} \in \mathcal{S}$. The 1-simplices are referred to as edges and the 2-simplices are called faces of the mesh. Two faces $\sigma_1, \sigma_2 \in \mathcal{S}$ are called adjacent if they meet in an edge, i. e., $\dim(\sigma_1 \cap \sigma_2) = 1$.

An example is depicted in Figure 1.6. For the goals pursued here, it is sufficient to consider meshes as structures which define shortest paths between points of the mesh (see Chapter 3) and specify adjacency and common borders of faces of the mesh (Chapter 5). Topological properties of meshes and simplicial complexes are studied in the area of computational topology [EH10], but are not crucial for the problems considered here.

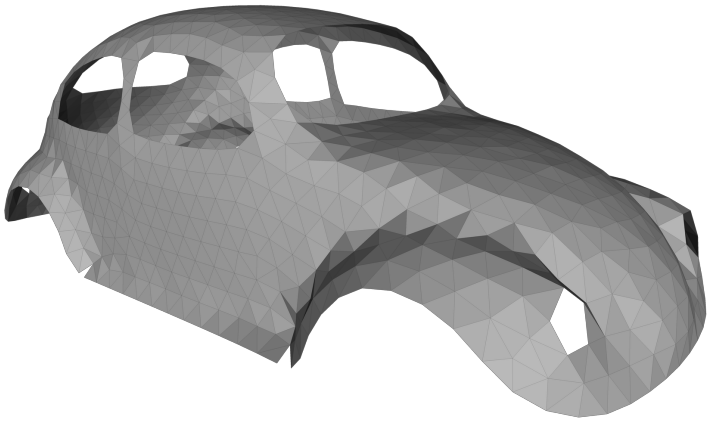


Figure 1.6: A mesh surface representing a car body[3].

A segmentation of a mesh is given by a subdivision of the mesh into submeshes.

Definition 1.10 (Mesh segmentation)
Let $\mathcal{S}$ be a mesh. A segmentation of $\mathcal{S}$ in $k \in \mathbb{N}$ submeshes is a subdivision $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_k = \mathcal{S}$ of $\mathcal{S}$ such that

- $\mathcal{S}_i$ is a mesh for all $i \in \{1, \ldots, k\}$
- $\sigma \in \mathcal{S}_i \cap \mathcal{S}_j \implies \dim(\sigma) \leq 1$ for all $i, j \in \{1, \ldots, k\}$, $i \neq j$,

i. e., each face of the mesh is contained in exactly one submesh $\mathcal{S}_i$, $i \in \{1, \ldots, k\}$.

Graphs and Ordered Sets

In the following chapters, we use ordered sets $\mathcal{S} \coloneqq (v_1, \ldots, v_t)$ to suitably describe certain algorithmic aspects. We use the common set operations and definitions on such ordered sets, e. g., we use $\mathcal{S} \subseteq V$ to denote that $\mathcal{S}$ only contains elements in V, and for $A \subseteq V$, $\mathcal{S} \setminus A$ is the ordered subset of $\mathcal{S}$ for which all $v \in A$ are deleted. Further, for $\mathcal{S} \subseteq \mathbb{R}^d$ the dimension of $\mathcal{S}$ is given by $\dim(\mathcal{S}) \coloneqq \dim(\{v_1, \ldots, v_t\})$. For a convenient notation, we write $\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$ to denote that we append the element v to the end of the ordered set.

An undirected graph with vertex set $V \coloneqq \{v_1, \ldots, v_n\}$ and a set of undirected edges $E \coloneqq \{e_1, \ldots, e_m\}$ is denoted as tuple $G \coloneqq (V, E)$. The edges are denoted as sets $\{v_i, v_j\} \in E$. To emphasize that a subgraph has exactly k vertices, we sometimes denote it as k-subgraph (e. g.,

[3]Figure generated from a surface mesh included in the matlab package "Toolbox Graph" by Gabriel Peyré.

a k-forest is a forest on exactly k vertices). The neighbourhood of a vertex v or a set of vertices A is referred to as $N_G(v)$ or $N_G(A)$. Here, $N_G(A) \coloneqq \{w \in V \setminus A \mid \exists v \in A \text{ with } \{v, w\} \in E\}$ does not contain any vertices in A. The set of vertices covered by a set of edges E' is denoted by $V(E')$, and $E(V')$ and $G(V') \coloneqq (V', E(V'))$ denote the edge set and subgraph induced by a vertex set V', respectively. For the sake of simplicity, we sometimes refer to subgraphs $(V(E'), E')$ by their inducing edge set E', and to subgraphs $(V', E(V'))$ by the inducing vertex set V'.

A graph can also be considered as metric space by using the shortest paths between vertices as distances.

Definition 1.11 (Shortest path distance)
Let $G \coloneqq (V, E)$ be a graph on $n \coloneqq |V|$ vertices. We denote by $d_G(v, w)$ the minimal number of edges of a path from v to w if v and w are connected, and $n + 1$ otherwise.

The shortest path distance defines a metric on a graph G and can be computed in polynomial time, see [Gri13].

In many cases, we consider weighted graphs $G \coloneqq (V, E, w)$, where $w\colon E \longrightarrow \mathbb{R}$ assigns a weight to each edge of the graph.

1.4 Acknowledgements

The interdisciplinary project which initiated the research for this thesis was an exciting opportunity to work on mathematical problems with a direct link to applications and practitioners. The corresponding grant *AIF 16895N* also partly funded my position at TUM. I am very grateful to my advisor Peter Gritzmann, not only for introducing me to this project and for many inspiring suggestions, but also for the freedom to develop and pursue my own ideas.

There are a lot of other people which have made this thesis possible. I would like to thank Michael Ritter for sharing his experience in the planning and execution of the cooperation with the project partners from engineering, for his guidance along the long path towards this thesis, and of course his patient technical support.

Steffen Borgwardt drew my attention to the dense k-subgraph problem, which resulted in our joint paper [BS14]. The fruitful and motivating discussions, not only on this particular topic, were a lot of fun.

Many thanks go also to Daniel Opritescu, who wrote his diploma on a mathematical aspect of the research project and later on joined the engineering side. His insight in both mechanical engineering as well as the mathematical challenges have played an invaluable role in leading the project to a very successful result.

René Brandenberg has guided my scientific attempts and teaching functions from the beginning. I am extremely grateful for his continuous support in mathematical questions, administrative tasks and many other things.

The working group at the chair for Discrete Mathematics and Applied Geometry has always provided an especially friendly and supportive atmosphere, and I am very grateful to all members for that. In particular, I would like to thank Andreas Alpers, Matthias Silbernagl and Stefan König for many insightful discussions on mathematics and all other things; Wolfgang Riedl, Viviana Ghiglione, Tobias Windisch, Andreas Alpers, Christian Böhm and Matthias Silbernagl provided many important suggestions for correcting and improving the manuscript.

Last, but not least, I would like to thank Marlene and my friends and family for their unconditional support.

Chapter 2

Affine Shape Matching

Recognizing special structures in certain objects is a key problem in computer science and in computer vision in particular. The applications for this task are manifold. The problem has been considered in computational biology, where one tries to match and classify molecule structures for synthetic drug design and biomolecular recognition [CB99] or in palaeobiology for the analysis of modern and ancient taxa [Mac99]. In computer vision, applications include the reconstruction of partially occluded objects [AMCO08], segmentation and sparse representation of point clouds [AFS06] and image analysis and segmentation in medical imaging [MV98].

Several approaches for this task have been considered, including matching local (e. g., curvature, characteristic curves [SS87], spin images [Joh97; JH99]) and global (eigenshapes [GT98], complex extended gaussian images [KI93], harmonic maps [ZH99]) features which are usually computed in a preprocessing step. A very active field of research is the alignment of point clouds, where one tries to find a transformation of a certain type (e. g., translation, Euclidean or affine transformation) which best aligns the point clouds with respect to a given measure. Countless algorithms have been proposed for problems of this type, where the works [BM92; GMO99; SWK07; AMCO08] seem to be the most relevant.

While plenty of measures for comparing matched point clouds have been considered, see [Vel01], the Hausdorff distance and the bottleneck distance are prevalent.

In this chapter, we present an analysis of a simple algorithm for this problem, which produces a transformation by aligning small subsets of both sets. The remaining part of the chapter is dedicated to adaptions of the algorithm to slightly modified problem settings. In Section 2.4, we describe a generalization of the algorithm for a multi-criteria optimization problem. Section 2.5 highlights a specific problem setting and tailor-made algorithm for a problem occurring in the application in mechanical engineering, cf. Chapter 1. The obtained results are partly based on joint work with Peter Gritzmann and Michael Ritter and currently in preparation for publication [GRS14].

2.1 The Largest Common Point Set Problem

For our application in engineering, it is unavoidable to allow only partial matching, i. e., we need to measure the amount by which the sets we try to align overlap. To assess the overlap of two point sets P and Q, several measures referred to as largest common point set (LCP) measure have been proposed for various settings. Akutsu, Tamaki, and Tokuyama [ATT98] give a definition with respect to congruence, which is generalized by Ambühl, Chakraborty, and Gärtner [ACG00] for approximate congruence. The largest common point set measure with respect to the set of Euclidean transformations $\mathcal{T}$ and the Hausdorff distance (Definition 1.2)

is given by

$$L_\delta(P,Q) \coloneqq \max\{|P'| \mid P' \subseteq P,\ Q' \subseteq Q \ \wedge\ \exists T \in \mathcal{T} \text{ with } d_H(T(P'),Q') \leq \delta\}. \tag{2.1}$$

Unfortunately, this measure is not meaningful if the set of admissible transformations $\mathcal{T}$ is extended to the set of affine transformations. The degenerate transformation $T\colon \mathbb{R}^d \longrightarrow \mathbb{R}^d$, $T(x) \coloneqq y$ for an arbitrary but fixed $y \in Q$, yields a largest common point set of cardinality $|P|$, irregardless of P and Q.

To avoid this problem, we use a slightly modified definition using relations which also allows us to describe the measure with respect to both the Hausdorff distance and the bottleneck distance with the same notation. At first, we consider the error of a relation with respect to a transformation. Throughout this chapter, $\|\cdot\|$ denotes an arbitrary norm on $\mathbb{R}^d$, cf. Section 1.3.

Definition 2.1 (Error of a relation)
Let $P,Q \subseteq \mathbb{R}^d$ be two sets, $T\colon \mathbb{R}^d \longrightarrow \mathbb{R}^d$ a transformation, and $R \subseteq P \times Q$ a relation between P and Q. We denote by

$$\Delta(R,T) \coloneqq \sup_{(p,q)\in R} \|T(p) - q\|$$

the error of R with respect to the transformation T.

The largest common point set measure can then be defined as follows:

Definition 2.2 (Largest common point set)
Let $P,Q \subseteq \mathbb{R}^d$ be finite point sets and $\delta \geq 0$. The δ-largest common point set measure for a set $\mathcal{T}$ of transformations and a set of relations $\mathcal{R}$ is given by

$$\mathcal{L}_\delta(P,Q) \coloneqq \max_{T\in\mathcal{T}} \max_{\substack{R\in\mathcal{R} \\ \Delta(R,T)\leq\delta}} \min\{|\mathcal{P}_1(R)|, |\mathcal{P}_2(R)|\}.$$

The transformations $T \in \mathcal{T}$ are called admissible transformations.

If the set of relations is unrestricted, i. e., $\mathcal{R} = 2^{P\times Q}$, we refer to the problem as δ-largest common point set problem with respect to the Hausdorff distance.

If $\mathcal{R}$ is the set of bottleneck relations (cf. Definition 1.4), $\mathcal{L}_\delta(P,Q)$ yields the δ-largest common point set with respect to the bottleneck distance.

Using the notation employed in the literature, the δ-largest common point set measure with respect to the Hausdorff distance can be expressed as

$$\mathcal{L}_\delta(P,Q) = \max\{\min\{|P'|,|Q'|\} \mid P' \subseteq P, Q' \subseteq Q' \ \wedge\ \exists T \in \mathcal{T} \text{ with } d_H(T(P'),Q') \leq \delta\}.$$

The corresponding decision problem for a given set of transformations $\mathcal{T}$ and relations $\mathcal{R}$ is the following:

Problem 2.3 (Largest common point set (LCP))
Input: Dimension $d \in \mathbb{N}$, point sets $P,Q \subseteq \mathbb{Q}^d, \delta \geq 0, L \in \mathbb{N}$.
Question: Is $\mathcal{L}_\delta(P,Q) \geq L$?

If one considers $\mathcal{T}$ to be the set of Euclidean transformations, it is known that the parameterized problem with the dimension d as parameter is $W[1]$-hard in L_2, even for $\delta = 0$ [CGK06]. This implies that it is unlikely that there exists an exact algorithm for this problem with

running time of the form $\mathcal{O}\Big(2^d \operatorname{poly}(|P|, |Q|)\Big)$; see [Nie06] for an introduction to parameterized complexity. More generally, the problem is shown to be $\mathcal{NP}$-hard in [ACG00] if the set of admissible transformations $\mathcal{T}$ is restricted to translations or linear, Euclidean, or affine transformations with respect to the bottleneck distance. The authors also present exact algorithms for various sets of admissible transformations. These include an exact algorithm for affine transformations in L_∞ with running time $(|P||Q|)^{2d^2+\mathcal{O}(d)}$, i. e., polynomial for fixed dimension d. However, this algorithm is computationally too expensive for our application in dimension $d = 3$. We therefore focus on an approximation algorithm, which yields very good results in practice.

2.2 Wide Bases

In the following, we analyse the approximation ratio of an elementary algorithm, which computes a transformation matching point sets P and Q based only on small (ordered) subsets — called *bases* — of both sets. Note that the approach is similar to a result by Akutsu [Aku96], who presents an algorithm and analysis for the special case considering Euclidean transformations in dimension $d = 3$ with respect to the Hausdorff distance, see Equation (2.1). Given P and Q, the algorithm finds subsets $P' \subseteq P$, $Q' \subseteq Q$ with $d_H(P', Q') \leq 8\delta$ such that P' is at least as large as the δ-largest common point set, i. e., $|P'| \geq \mathcal{L}_\delta(P, Q)$. A modified algorithm for this particular problem setting with improved approximation ratio 2 is given in [CB99].

In this chapter, we present results on Problem LCP where the admissible transformations are the set of affine transformations

$$\mathcal{A} \coloneqq \Big\{T\colon \mathbb{R}^d \longrightarrow \mathbb{R}^d \mid T(x) = Ax + t, A \in \mathbb{R}^{d\times d}, t \in \mathbb{R}^d\Big\}$$

in arbitrary dimension. Note that degenerate transformations, i. e., with $\operatorname{rank}(A) < d$, are not excluded. In the following algorithm, we utilize the fact that affine transformations can be defined by appropriate ordered sets:

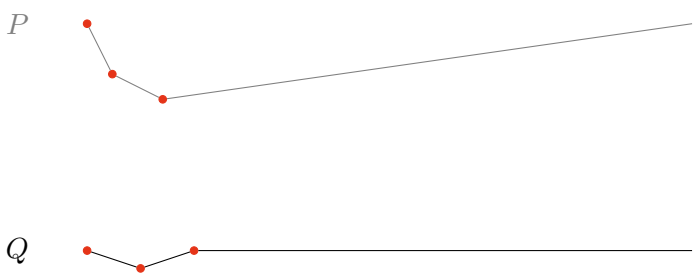
Definition 2.4
Let $B_P \coloneqq (p_0, \ldots, p_n) \subseteq \mathbb{R}^d$ be affinely independent and $B_Q \coloneqq (q_0, \ldots, q_n) \subseteq \mathbb{R}^d$ an ordered set. We denote by $T_{B_P,B_Q} \in \mathcal{A}$ an arbitrary affine transformation such that

$$T_{B_P,B_Q}(p_i) = q_i \quad \forall i \in \{0, \ldots, n\}.$$

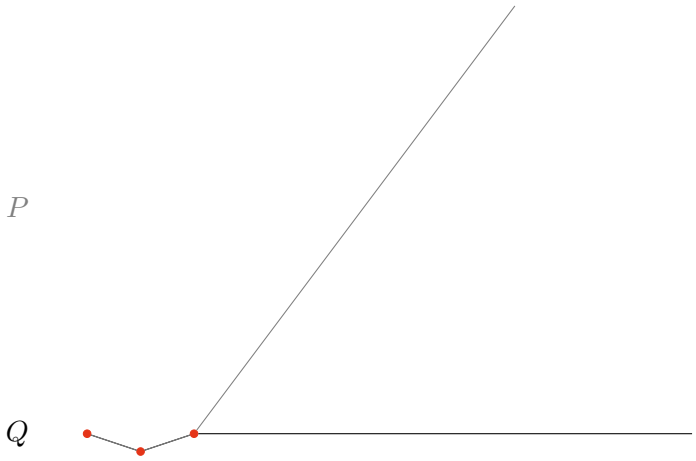
The existence of such a transformation is a basic result from linear algebra.

Remark 2.5
Denoting $M_P \coloneqq (p_1 - p_0, \ldots, p_n - p_0) \in \mathbb{R}^{d\times n}$ and $M_Q \coloneqq (q_1 - q_0, \ldots, q_n - q_0) \in \mathbb{R}^{d\times n}$, a transformation as in Definition 2.4 is given by $T_{B_P,B_Q}(x) \coloneqq Ax + t$ where $A \in \mathbb{R}^{d\times d}$ satisfies $AM_P = M_Q$ and $t \coloneqq q_0 - Ap_0$. The transformation is unique if $\dim(B_P) = d$, i. e., $n = d$. In this case we have $A = M_Q M_P^{-1}$.

While it is obvious that a transformation which matches arbitrary ordered subsets of P and Q may yield an arbitrarily bad alignment of P and Q in general, cf. Figure 2.1, the error can be bounded for bases with certain properties. Intuitively, it is clear that the points of a base should be far apart in order to minimize the influence of noise. This has already been



(a) Point sets P and Q which can be aligned with low error by a translation.



(b) Point sets $T_{B_P,B_Q}(P)$ and Q for the transformation defined by the bases B_P, B_Q.

Figure 2.1: Unfavourable choice of bases B_P, B_Q (depicted in red) of P (grey) and Q (black).

mentioned in [AMCO08] and appropriate sets are referred to as a *wide bases*. We introduce a rigorous characterization of this term in Definition 2.6.

The algorithm can then be stated in the form of Algorithm 2.1. For all pairs of bases of P and Q, a transformation T matching these bases and a corresponding relation are computed and evaluated. Note that

$$R := \{(p, q) \in P \times Q \mid \|T(p) - q\| \leq \delta\} \subseteq P \times Q$$

is an optimal relation with respect to the Hausdorff distance. An optimal bottleneck relation can be obtained by solving a maximal bipartite matching problem. The best of these relations and the corresponding transformation is then returned.

The fact that this algorithm computes an approximation of an optimal solution is based on two key properties: A transformation obtained by matching a wide base of P with an appropriate base of Q is an approximation of the transformation best aligning these sets, see Theorem 2.13. Furthermore, Theorem 2.7 guarantees the existence of such wide bases.

The running time of the algorithm is dominated by the $\mathcal{O}\Big(d(|P||Q|)^{d+1}\Big)$ loop iterations for the elementary version stated in Algorithm 2.1. In practice, more sophisticated methods can be used to drastically improve the running time of the algorithm. This is usually achieved by heuristically disregarding certain bases, while sacrificing quality of approximation. For

Input: $P, Q \subseteq \mathbb{R}^d$ finite point sets, $\delta \geq 0$
Output: Transformation T, relation R

$L \leftarrow -1$
for $n = 0$ *to* d **do**
  for *all affinely independent ordered sets* $B_P \subseteq P$ *with* $|B_P| = n+1$ *and all ordered multisets* $B_Q \subseteq Q$ *with* $|B_Q| = n+1$ **do**
    $T' \leftarrow T_{B_P, B_Q}$
    $R' \leftarrow \operatorname{argmax}\{\min\{|\mathcal{P}_1(R')|, |\mathcal{P}_2(R')|\} \mid R' \in \mathcal{R},\ \Delta(R', T') \leq \delta\}$
    if $\min\{|\mathcal{P}_1(R')|, |\mathcal{P}_2(R')|\} > L$ **then**
      $L \leftarrow \min\{|\mathcal{P}_1(R')|, |\mathcal{P}_2(R')|\}$
      $T \leftarrow T'$
      $R \leftarrow R'$
    end
  end
end

Algorithm 2.1: affineBaseMatching

the problem considering Euclidean transformations in dimension $d = 3$, Schnabel, Wahl, and Klein [SWK07] use probabilistic methods to significantly reduce the number of bases which are considered. Aiger, Mitra, and Cohen-Or [AMCO08] use slightly refined bases, which allow for a further reduction of pairs of bases which have to be evaluated. These concepts can also be transferred to the problem concerning affine transformations, see Section 2.5.

In the following discussion, it turns out that, in the case of affine transformations, a suitable formal definition of a wide base $B \coloneqq \{b_0, \dots, b_d\}$ of P is the following: after choosing a centre $b_0 \in P$, each point of P can be expressed as an affine combination of vectors in the base, such that the absolute values of the coefficients of $b_1, \dots, b_d$ are less than or equal to 1. This is equivalent to the statement that P is contained in the parallelotope $b_0 + \sum_{b \in B} [-1, 1](b - b_0)$ centred in b_0 and spanned by the vectors $b - b_0$ for $b \in B$.

Definition 2.6 (Wide base)
Let $P \subseteq \mathbb{R}^d$. We call a subset $B \subseteq P$ with $|B| = \dim(P) + 1$ wide base of P if there exists $b_0 \in B$ such that

$$P - b_0 \subseteq \operatorname{par}(B - b_0) \coloneqq \left\{ \sum_{b \in B} \lambda_b (b - b_0) \mid \lambda_b \in \mathbb{R},\ |\lambda_b| \leq 1\ \forall b \in B \right\} = \sum_{b \in B} [-1, 1](b - b_0).$$

The point b_0 is called centre and the vectors $b - b_0$ are called basis of the parallelotope $\operatorname{par}(B - b_0)$ or B. We refer to B as ordered wide base if B is an ordered subset of P.

For an example, consider the circle $P \coloneqq \{x \in \mathbb{R}^2 \mid \|x - e_1\| = 1\}$, cf. Figure 2.2, and $b_0 \coloneqq 0$. Then the regular simplex $B \coloneqq \{b_0, b_1, b_2\}$ where $b_1 \coloneqq \left(\frac{3}{2}, \frac{\sqrt{3}}{2}\right)^T$ and $b_2 \coloneqq \left(\frac{3}{2}, -\frac{\sqrt{3}}{2}\right)^T$ is a wide base of P containing b_0.

An essential result for our analysis of the approximation ratio of Algorithm 2.1 is the following, which states that each finite point set contains such a wide base.

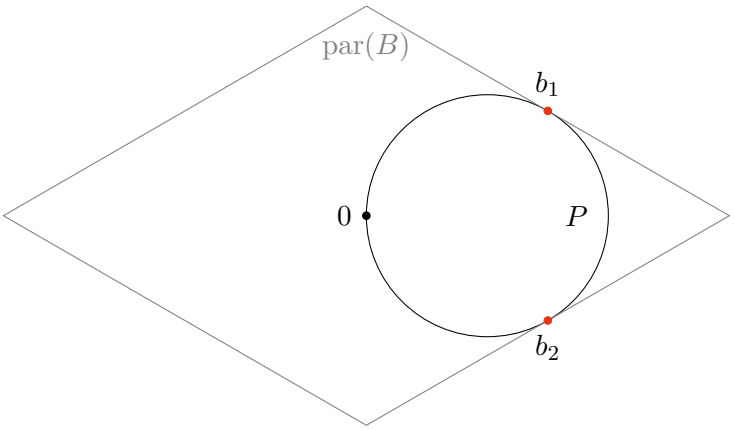


Figure 2.2: Wide base of a unit circle.

Theorem 2.7
For each compact set $P \subseteq \mathbb{R}^d$, $n \coloneqq \dim(P)$, and each $b_0 \in P$, there are $b_1, \ldots, b_n \in P$ such that $\{b_0, \ldots, b_n\}$ is a wide base of P with centre b_0.

We use a supporting lemma to prove Theorem 2.7, but first we show that the restriction to compact sets is indeed necessary.

Remark 2.8
The set P obviously has to be bounded, since it could not be contained in a bounded parallelotope otherwise.

If P is not closed, Theorem 2.7 does not hold in general: Consider the open interval $P \coloneqq (0,1) = \{x \in \mathbb{R} \mid 0 < x < 1\}$ and assume $B \coloneqq \{b_0, b_1\} \subseteq P$ is a wide base with centre b_0. We have

$$\operatorname{par}(B - b_0) = [-|b_1 - b_0|, |b_1 - b_0|] = \{x \in \mathbb{R} \mid \ -|b_1 - b_0| \leq x \leq |b_1 - b_0|\}$$

and $P - b_0 = (-b_0, 1 - b_0)$. If $b_0 \leq b_1$, we have $1 - b_0 > b_1 - b_0 = |b_1 - b_0|$. For $b_0 > b_1$, it holds $-b_0 < -b_0 + b_1 = -|b_1 - b_0|$. Combined, this yields $P - b_0 \not\subseteq \operatorname{par}(B - b_0)$, and therefore P does not contain a wide base.

The following lemma states that each full-dimensional compact set P contains d points which span a parallelotope that contains the set P itself.

Lemma 2.9
Let $P \subseteq \mathbb{R}^d$ be compact and full-dimensional. P contains a linearly independent subset $B \subseteq P$ with $|B| = \dim(P) = d$ such that

$$P \subseteq \operatorname{par}(B) = \sum_{b \in B} [-1, 1]b.$$

Proof. Let $\det(b_1, \ldots, b_d)$ denote the determinant of the matrix consisting of the columns $b_1, \ldots, b_d \in \mathbb{R}^d$. Since P is compact and the volume $\operatorname{vol}_p \colon \mathbb{R}^d \times \ldots \times \mathbb{R}^d \longrightarrow \mathbb{R}_{\geq 0}$,

$$\operatorname{vol}_p(b_1, \ldots, b_d) \coloneqq \operatorname{vol}(\operatorname{par}(b_1, \ldots, b_d)) = 2^d |\det(b_1, \ldots, b_d)|$$

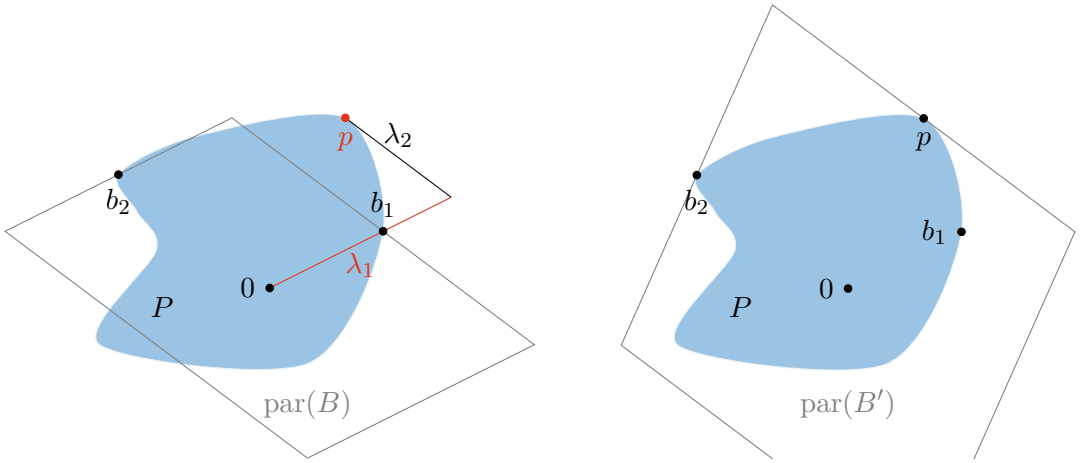
is a continuous function, there exists a set $B \coloneqq \{b_1, \ldots, b_d\} \subseteq P$ of d points in P which span a parallelogram of maximal volume:

$$\operatorname{vol}(\operatorname{par}(B)) = \max_{\substack{B' \subseteq P \\ |B'|=d}} \operatorname{vol}(\operatorname{par}(B')).$$

Since $\dim(P) = d$, we have $\operatorname{vol}(\operatorname{par}(B)) > 0$, i. e., B is linearly independent. Assume now that there exists $p \in P \setminus \operatorname{par}(B) \neq \emptyset$. There are $\lambda_1, \ldots, \lambda_d \in \mathbb{R}$ such that

$$p = \sum_{i=1}^{d} \lambda_i b_i$$

and we can assume $|\lambda_1| \geq \ldots \geq |\lambda_d|$ without loss of generality. Since $p \in P \setminus \operatorname{par}(B)$, we have $|\lambda_1| > 1$. This situation is illustrated in Figure 2.3a.



(a) A linear combination $p \in P \setminus \operatorname{par}(B)$ of vectors $b \in B$ must have at least one coefficient $|\lambda_1| > 0$.

(b) The volume of the parallelotope can be increased by exchanging b_1 and p: $B' \coloneqq (B \setminus \{b_1\}) \cup \{p\}$.

Figure 2.3: Illustration of the proof of Lemma 2.9.

Consider $A \coloneqq (B \setminus \{b_1\})$ and let $u \in A^{\perp} \neq \emptyset$, $\|u\| = 1$ be a unit normal of the linear subspace spanned by A. We have

$$\begin{aligned}
|\langle u, p \rangle| &= \left| \left\langle u, \sum_{i=1}^{d} \lambda_i b_i \right\rangle \right| \\
&= \left| \lambda_1 \langle u, b_1 \rangle + \sum_{i=2}^{d} \lambda_i \underbrace{\langle u, b_i \rangle}_{=0} \right| \\
&= |\lambda_1| |\langle u, b_1 \rangle| \\
&> |\langle u, b_1 \rangle|.
\end{aligned}$$

Let $\mathrm{vol}_{\mathrm{aff}(A)}$ denote the volume in the affine subspace $\mathrm{aff}(A)$. For $B' \coloneqq A \cup \{p\}$ we have

$$\begin{aligned}\mathrm{vol}(\mathrm{par}(B')) &= 2|\langle u, p\rangle| \cdot \mathrm{vol}_{\mathrm{aff}(A)}(\mathrm{par}(A)) \\ &> 2|\langle u, b_1\rangle| \cdot \mathrm{vol}_{\mathrm{aff}(A)}(\mathrm{par}(A)) \\ &= \mathrm{vol}(\mathrm{par}(B)),\end{aligned}$$

cf. Figure 2.3b. This is a contradiction to the maximality of the parallelogram spanned by B, implying that indeed $P \subseteq \mathrm{par}(B)$. □

This proof also establishes that a parallelotope with centre 0 and basis in P of maximal volume contains all points of P.

Corollary 2.10
Let $P \subseteq \mathbb{R}^d$ and $B \subseteq P$ be the basis of a parallelotope with maximal volume, then $P \subseteq \mathrm{par}(B)$.

Note that the opposite statement does not hold in general. Consider, for example, the point set $P \coloneqq \{e_1, e_2, e_1 + e_2, -e_1 + e_2\} \subseteq \mathbb{R}^2$. We obviously have $P \subseteq \mathrm{par}(\{e_1, e_2\})$ and also $P \subseteq \mathrm{par}(\{e_1 + e_2, -e_1 + e_2\})$, but $\mathrm{vol}(\mathrm{par}(\{e_1 + e_2, -e_1 + e_2\})) = 8 > 4 = \mathrm{vol}(\mathrm{par}(\{e_1, e_2\}))$, compare Figure 2.4.

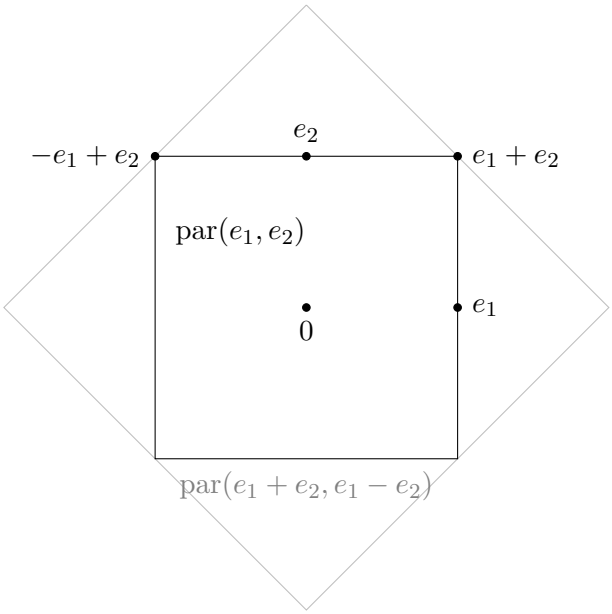


Figure 2.4: Wide bases need not induce parallelotopes of maximal volume.

It is now straightforward to prove Theorem 2.7, by a suitable translation of the point set P and Lemma 2.9.

Proof (Theorem 2.7). For $P \subseteq \mathbb{R}^d$, $n \coloneqq \dim(P)$ and $b_0 \in P$, consider the point set $P' \coloneqq \{p - b_0 \mid p \in P\} \subseteq \mathbb{R}^d$. We can consider P' as a full-dimensional set in the linear subspace spanned by P'. In this setting, Lemma 2.9 states that there are $b'_1, \ldots, b'_n \in P'$ such that $P' \subseteq \mathrm{par}(b'_1, \ldots, b'_n)$. Thus, $B \coloneqq \{b_0, b'_1 + b_0, \ldots, b'_n + b_0\}$ is a wide base of P. This concludes the proof of Theorem 2.7. □

2.2.1 Finding a wide base

While the existence of wide bases will prove sufficient for the bound on the approximation ratio of Algorithm 2.1, the computation of a wide base for a given finite point set is an interesting problem in its own right. The problem is loosely related to containment problems, cf. [GK94], but does not belong to any of the categories described in the survey.

Considering Corollary 2.10, an intuitive idea for computing a wide base with centre b_0 of a point set $P \subseteq \mathbb{R}^d$ would be to determine a basis B which induces a parallelotope of maximal volume

$$\max_{\substack{B \subseteq P - b_0 \\ |B| = d}} \operatorname{vol}(\operatorname{par}(B)).$$

This more general problem is similar to the problem of computing the volume of the zonotope $Z \coloneqq \sum_{p \in P} [-1, 1](p - b_0)$ spanned by $P - b_0$, which can be stated as

$$\sum_{\substack{B \subseteq P - b_0 \\ |B| = d}} \operatorname{vol}(\operatorname{par}(B)),$$

and is $\#\mathcal{P}$-hard, see [DGH98]. However, the proof can not easily be transferred to the problem of computing a parallelotope of maximal volume. To the best of our knowledge, no results on the computational complexity of computing a basis B inducing a polytope of maximal volume or determining a wide base can be found in the literature.

However, the proofs of Lemma 2.9 and Theorem 2.7 also disclose an algorithmic idea for constructing a wide base.

We start with an arbitrary (affinely independent) base. If there exists a point $p \in P$ which is not contained in the parallelotope corresponding to the current base, the volume of the parallelotope can be increased by exchanging an element of the current base with p. This procedure is iterated until a wide base is found. Since there is only a finite number of possible bases and each base is visited at most once (the volume of the parallelotope increases in each iteration), the procedure computes a wide base in finitely many iterations.

To formally describe this algorithm, we investigate representations which are more suitable for this purpose.

Remark 2.11
For a finite point set $P \coloneqq \{p_1, \ldots, p_k\} \subseteq \mathbb{R}^d$ with $\dim(P) = d$, Theorem 2.7 can also be stated as follows: For $i_0 \in \{1, \ldots, k\}$, let $A \coloneqq (p_1 - p_{i_0}, \ldots, p_k - p_{i_0}) \in \mathbb{R}^{d \times k}$. There are mutually distinct indices $B \coloneqq \{i_1, \ldots, i_d\} \subseteq \{1, \ldots, k\} \setminus \{i_0\}$, such that the matrix $A_B \coloneqq (p_{i_1} - p_{i_0}, \ldots, p_{i_d} - p_{i_0}) \in \mathbb{R}^{d \times d}$ consisting of the columns $i_1, \ldots, i_d$ of A is non-singular, and we have $\left| \left(A_B^{-1} A \right)_{r,s} \right| \leq 1$ for all $r \in \{1, \ldots, d\}$, $s \in \{1, \ldots, k\}$.

Algorithm 2.2 implements the algorithmic idea described above, using the notation introduced in Remark 2.11. Note that this algorithm is very similar to the tableau form of the dual simplex algorithm, the only difference being the pivoting rule. The algorithm can therefore be implemented in the same very efficient way as the simplex tableau. Since $A_B^{-1} A$ is known and $|B \triangle B'| = 2$, $A_{B'}^{-1} A$ can be computed with d row operations after the first iteration.

Another way to see the similarity to the simplex algorithm is to consider the procedure of Algorithm 2.2 as an edge walk on the matroid base polytope I_P associated with the matrix A:

$$I_P \coloneqq \operatorname{conv}\Big(\Big\{ e \in \{0,1\}^k \mid \mathbb{1}_d^T e = d \ \wedge \ \{p_i - p_0 \mid e_i = 1\} \text{ is linearly independent} \Big\} \Big).$$

Input: $P \coloneqq \{p_1, \ldots, p_k\} \subseteq \mathbb{R}^d$ with $\dim(P) = d$, indices of initial (non-wide) base $B \coloneqq \{i_0, \ldots, i_d\} \subseteq \{1, \ldots, k\}$
Output: Indices of wide base B

$A \leftarrow (p_1 - p_{i_0}, \ldots, p_k - p_{i_0})$
repeat
  Find indices (r, s) of maximal element of $\left|A_B^{-1} A\right|$
  $m \leftarrow \left|(A_B^{-1} A)_{r,s}\right|$
  $B \leftarrow (B \setminus \{i_r\}) \cup \{s\}$
until $m \leq 1$

Algorithm 2.2: constructWideBase

There is a one-to-one correspondence between the vertices of this polytope and the bases B of A: two bases B, B' are connected by an edge of the polytope if and only if they differ in exactly one element, i. e., $|B \triangle B'| = 2$. This is a direct consequence of Theorem 40.6 and Corollary 40.2d of [Sch03]. The coefficients of a linear combination $p_s - p_0$ of $p_{i_r} - p_0$, $i_r \in B$, can be considered as directed edge weights: the weight of the directed edge from B to B' with $B' = B \setminus \{i_r\} \cup \{s\}$ for $r \in \{1, \ldots, d\}, s \in \{1, \ldots, k\}$ is given by $(A_B^{-1} A)_{r,s}$.

The diameter of this polytope is bounded from above by d, since it is a 0-1-polytope, cf. [Nad89]. This bound can also be concluded directly using Steinitz' exchange lemma [Str03]. Combined with computational experiments, this motivates the following conjecture:

Conjecture 2.12
During a run of Algorithm 2.2, each element may enter the base at most $d-1$ times. Thus, at most $(d-1)n$ loop iterations are performed.

2.3 An Approximation Algorithm

We are now ready to prove the main result of this chapter, namely that aligning a wide base of P with a suitable base of Q yields an approximation for the best alignment of these sets.

Theorem 2.13
Let $P, Q \subseteq \mathbb{R}^d$ be compact sets, $T \in \mathcal{A}$ an affine transformation, $R \subseteq P \times Q$ a relation, and $n \coloneqq \dim(\mathcal{P}_1(R))$. For each ordered wide base B_P of $\mathcal{P}_1(R)$, there exists an ordered multiset $B_Q \subseteq \mathcal{P}_2(R) \subseteq Q$ with $|B_Q| = n + 1 = |B_P|$ such that

$$\Delta\Big(R, T_{B_P, B_Q}\Big) \leq 2(n+1)\Delta(R, T)$$

for the affine transformation T_{B_P,B_Q} introduced in Definition 2.4.

Proof. Let $b_0, \ldots, b_n \in P$ be such that $B_P = (b_0, \ldots, b_n)$ and $\Delta \coloneqq \Delta(R, T)$. There exists an optimal aligning transformation consisting of $A_o \in \mathbb{R}^{d \times d}, t_o \in \mathbb{R}^d$ such that

$$\|A_o p + t_o - q\| \leq \Delta \qquad \forall (p, q) \in R.$$

Let $q_0, \ldots, q_n \in Q$ be such that $(b_i, q_i) \in R, i \in \{0, \ldots, n\}$ and denote $B_Q \coloneqq (q_0, \ldots, q_n)$.

Since B_P is affinely independent, there exist $A \in \mathbb{R}^{d\times d}$ and $t \in \mathbb{R}^d$ with

$$\|Ab_i + t - q_i\| = 0 \qquad \forall i \in \{0, \dots, n\},$$

i. e., T_{B_P,B_Q} consists of the linear transformation A and translation t.

Let $c \in \{0, \dots, n\}$ be such that b_c is a centre of the wide base B_P, cf. Definition 2.6. We can assume $b_c = 0$ without loss of generality by translating P. Since $b_c = 0$, we have $q_c = Ab_c + t = 0 + t = t$ and therefore

$$\begin{aligned}
\|Ap + t - q\| &= \|Ap + q_c - q\| \\
&\le \|A_o p - (q - q_c)\| + \|(A - A_o)p\| \\
&= \|A_o p + t_o - q - (t_o - q_c)\| + \|(A - A_o)p\| \\
&\le \|A_o p + t_o - q\| + \|A_o b_c + t_o - q_c\| + \|(A - A_o)p\| \\
&\le 2\Delta + \|(A - A_o)p\|
\end{aligned}$$

for all $(p, q) \in R$. Thus, the distance between the transformation of the point p and q is bounded by a term of 2Δ plus the deviation of the linear transformation.

Next, we note that the error of the linear transformation of the points $b_i \in B_P$ in the base, $i \in \{0, \dots, n\}$, is bounded by 2Δ: since $q_c = t$ and $(b_c, q_c) \in R$, we have $\|t - t_o\| \le \Delta$ and therefore

$$\begin{aligned}
\|(A - A_o)b_i\| &\le \|(Ab_i + t) - (A_o b_i + t_o)\| + \|t - t_o\| \\
&\le \|q_i - (A_o b_i + t_o)\| + \Delta \\
&\le 2\Delta.
\end{aligned}$$

Since B_P is a wide base of $\mathcal{P}_1(R)$ with centre $b_c = 0$, each $p \in \mathcal{P}_1(R)$ can be represented as

$$p = \sum_{i\neq c} \lambda_p^i b_i \text{ with } \left|\lambda_p^i\right| \le 1$$

for suitable $\lambda_p^i \in \mathbb{R}$, $i \in \{0, \dots, n\} \setminus \{c\}$. The error of the linear transformation of $p \in \mathcal{P}_1(R)$ can now be bounded as follows:

$$\begin{aligned}
\|(A - A_o)p\| &= \left\|(A - A_o)\left(\sum_{i\neq c} \lambda_p^i b_i\right)\right\| \\
&\le \sum_{i\neq c} \left|\lambda_p^i\right| \|(A - A_o)b_i\| \\
&\le \sum_{i\neq c} \left|\lambda_p^i\right| 2\Delta \\
&\le 2n\Delta.
\end{aligned}$$

□

Substituting in the above result yields $\|Ap + t - q\| \le 2(n + 1)\Delta$ for all $(p, q) \in R$.

Furthermore, the bound on the approximation ratio stated in Theorem 2.13 is tight.

Lemma 2.14
For each $d \in \mathbb{N}$, there exist finite point sets $P \coloneqq \{p_0, \dots, p_{d+1}\}$, $Q \coloneqq \{q_0, \dots, q_{d+1}\} \subseteq \mathbb{R}^d$ with corresponding wide bases B_P, B_Q such that the affine transformation T_{B_P,B_Q} defined in Definition 2.4 fulfils

$$\Delta\Big(R, T_{B_P,B_Q}\Big) = 2(d+1)\Delta(R,T)$$

for the identity transformation $T \coloneqq id$ and correspondence $R \coloneqq \{(p_i, q_i) \mid i \in \{0, \dots, d+1\}\}$.

Proof. Let $P \coloneqq \{p_0, \dots, p_{d+1}\}$, where $p_0 \coloneqq 0$, $p_i \coloneqq e_i$ for $i \in \{1, \dots, d\}$ and $p_{d+1} \coloneqq -\mathbb{1}_d$. Similarly, for $\delta \geq 0$, we set $Q \coloneqq \{q_0, \dots, q_{d+1}\}$, where $q_0 \coloneqq \delta e_1$, $q_i \coloneqq -\delta e_1 + e_i$ for $i \in \{1, \dots, d\}$ and $q_{d+1} \coloneqq -\mathbb{1}_d - \delta e_1$. We obviously have $\|p_i - q_i\| \leq \delta$ for all $i \in \{0, \dots, d+1\}$ and therefore $\Delta(R, id) \leq \delta$. Furthermore $B_P \coloneqq \{p_0, \dots, p_d\}$ is a wide base of P, and for $B_Q \coloneqq \{q_0, \dots, q_d\}$, the corresponding affine transformation $T \coloneqq T_{B_P,B_Q}$ is given by $T(x) = Ax + t$, where

$$A \coloneqq \begin{pmatrix} 1-2\delta & -2\delta \mathbb{1}_{d-1}^T \\ 0 & I_{d-1} \end{pmatrix}, \qquad t \coloneqq \delta e_1.$$

Further, we have

$$\begin{aligned} T(p_{d+1}) = t + Ap_{d+1} &= \delta e_1 + (-1 + 2\delta + (d-1)2\delta)e_1 - \sum_{i=2}^{d} e_i \\ &= -\mathbb{1}_d + (2d+1)\delta e_1 \end{aligned}$$

which yields

$$\|T(p_{d+1}) - q_{d+1}\| = 2(d+1)\delta.$$

For sufficiently small δ we have

$$\|T(p_{d+1}) - q_i\| \geq \sqrt{d-2} \geq 2(d+1)\delta$$

for all $i \in \{1, \dots, d\}$ and therefore $\Delta\Big(R, T_{B_P,B_Q}\Big) \geq 2(d+1)\delta$. □

The obtained results can now be combined to establish a bound on the approximation ratio of Algorithm 2.1.

Corollary 2.15
For $\Delta \geq 0$ and two finite sets $P, Q \subseteq \mathbb{R}^d$, let $\delta \coloneqq 2(d+1)\Delta$. Algorithm 2.1 finds a relation $R \in \mathcal{R}$ and a transformation $T \in \mathcal{A}$ such that

$$\Delta(R,T) \leq \delta = 2(d+1)\Delta \text{ and } \min\{|\mathcal{P}_1(R)|, |\mathcal{P}_2(R)|\} \geq \mathcal{L}_\Delta(P,Q).$$

Proof. Let $T_o \in \mathcal{A}$ be an optimal transformation and $R_o \subseteq P \times Q$ a corresponding optimal relation with respect to the LCP measure. By Theorem 2.7, there exists an ordered wide base $B_{P'} \coloneqq (b_0, \dots, b_n)$ of $P' \coloneqq \mathcal{P}_1(R) \subseteq P$, where $n \coloneqq \dim(P')$. Theorem 2.13 states that there exists an ordered set $B_{Q'} \subseteq \mathcal{P}_2(R) \subseteq Q$, $|B_{Q'}| = n$ such that $\Delta\Big(R_o, T_{B_{P'},B_{Q'}}\Big) \leq 2(d+1)\Delta$ for an affine transformation $T_{B_{P'},B_{Q'}}$ matching $B_{P'}$ to $B_{Q'}$. Since Algorithm 2.1 iterates over all ordered multisets of up to $d+1$ points, the transformation matching $B_{P'}$ to $B_{Q'}$ is considered in the course of the algorithm. □

The transformation computed by Algorithm 2.1 is not necessarily a local optimum, since at most $d+1$ points were considered. In practical applications, it is therefore reasonable to perform a local optimization of the transformation. In L_2, this is possible in polynomial time.

Lemma 2.16
Let $P, Q \subseteq \mathbb{R}^d$ be finite point sets and $R \subseteq P \times Q$ a relation. The problem

$$\min_{\substack{A\in\mathbb{R}^{d\times d}\\ t\in\mathbb{R}^d}} \max_{(p,q)\in R} \|q-(Ap+t)\|_2$$

can be approximated to any accuracy in polynomial time.

Proof. The problem can be formulated as the following second order cone program (SOCP), and therefore approximated to any accuracy in polynomial time, cf. [MT00].

$$\begin{array}{lrcll} \min & \delta & & & \\ \text{s.t.} & \|q-(Ap+t)\|_2 & \leq & \delta & \forall (p,q)\in R \\ & A & \in & \mathbb{R}^{d\times d} & \\ & t & \in & \mathbb{R}^d & \end{array}$$

This is indeed an SOCP: Let $p_1,\ldots,p_k \in P$, $q_1,\ldots,q_k \in Q$ be (not necessarily distinct) points such that $R=\{(p_i,q_i) \mid i=\{1,\ldots,k\}\}$, $I_d \in \mathbb{R}^{d\times d}$ the identity matrix,

$$M_i := \begin{pmatrix} p_i^T & 0 & \ldots & 0 & & 0 \\ 0 & p_i^T & \ddots & \vdots & I_d & \vdots \\ \vdots & \ddots & \ddots & 0 & & \\ 0 & \ldots & 0 & p_i^T & & 0 \end{pmatrix} \in \mathbb{R}^{d\times(d(d+1)+1)},$$

$$f := c_i := e_{d(d+1)+1} \in \mathbb{R}^{d(d+1)+1}$$

and

$$b_i := -q_i \in \mathbb{R}^d$$

for $i \in \{1,\ldots,k\}$. With this notation, the above program is equivalent to the following SOCP in standard notation:

$$\begin{array}{lrcll} \min & f^T x & & & \\ \text{s.t.} & \|M_i x + b_i\|_2 & \leq & c_i^T x & \forall i\in\{1,\ldots,k\} \\ & x & \in & \mathbb{R}^{d(d+1)+1} & \end{array}$$

where A and t and δ are given by

$$A := \begin{pmatrix} x_1 & \ldots & x_d \\ \vdots & \ddots & \vdots \\ x_{d(d-1)+1} & \ldots & x_{d^2} \end{pmatrix} \in \mathbb{R}^{d\times d}, \qquad t := \begin{pmatrix} x_{d^2+1} \\ \vdots \\ x_{d(d+1)} \end{pmatrix} \in \mathbb{R}^d$$

and $\delta := x_{d(d+1)+1}$. □

This process can of course be iterated in the same manner used in the classical ICP-Algorithm, cf. [BM92]:

For a given relation, we use the overlapping point sets $P' := \mathcal{P}_1(R) \subseteq P$ and $Q' := \mathcal{P}_2(R) \subseteq Q$ to compute a transformation T which aligns these sets as described in Lemma 2.16. A relation R' which maximizes $\min\{\mathcal{P}_1(R), \mathcal{P}_2(R)\}$ under the constraint $\Delta(R,T) \leq \delta$ can then be computed. This process yields the overlapping point sets $P'' := \mathcal{P}_1(R') \supseteq P'$ and $Q'' := \mathcal{P}_2(R') \supseteq Q'$, and the process can be iterated.

2.4 Pareto Approximation

In some applications, there is no canonical bound δ on the maximal acceptable error of a relation. It is therefore desirable to strike a good balance between the opposing goals of covering a big proportion of the sets and obtaining a relation with small error. Instead of looking for maximal subsets of P and Q which can be transformed such that their Hausdorff distance is below a threshold δ, we can also consider the problem as a multi-criteria or Pareto optimization problem.

Definition 2.17 (Multi-criteria optimization)
Let D be a set. A map $f\colon D \longrightarrow \mathbb{R}^n$ is called multi-criteria objective function, where n is the number of criteria considered.

Since a multi-criteria optimization problem usually does not have a single distinguished optimal point, we are interested in the set of points where no individual criterion can be improved without impairing other criteria. This set of points is usually referred to as Pareto front. As we have seen in the introduction to this chapter, even the problem with a one-criterion objective function is very involved, so the computation of the exact Pareto front is not feasible for practical instances. Instead, we investigate an approximation.

Definition 2.18 (Approximate Pareto front)
Let D be a set, $f\colon D \longrightarrow \mathbb{R}^n$ a multi-criteria objective function, and $\alpha \in \mathbb{R}^n_{\geq 0}$. A subset $\mathcal{F} \subseteq D$ is called α-approximate Pareto front for the minimization of f if

i) for all $q \in D$ there exists $p \in \mathcal{F}$ such that $f_i(p) \leq \alpha_i f_i(q)$ for all $i \in \{1, \ldots, n\}$

ii) $f(p_1) \not\leq f(p_2)$ for all $p_1, p_2 \in \mathcal{F}$.

An $\mathbb{1}_d$-approximate Pareto front is called (exact) Pareto front.

The canonical criteria which determine the quality of a relation $R \subseteq X \times Y$ and corresponding transformation T are the number of points in X and Y which are covered by R, i. e., $\mathcal{P}_1(R)$ and $\mathcal{P}_2(R)$, and the error $\Delta(R, T)$ induced by R and T. A reasonable multi-criteria objective function for the largest common point set problem is given in the following definition:

Definition 2.19 (Multi-criteria largest common point set)
Let $P, Q \subseteq \mathbb{R}^d$ be two finite point sets, $\mathcal{R}$ a set of relations of P and Q, and $\mathcal{T}$ a set of admissible transformations of $\mathbb{R}^d$. We define the multi-criteria objective function $f\colon \mathcal{R} \times \mathcal{T} \longrightarrow \mathbb{R}^3$ by

$$f(R, T) \coloneqq (-\mathcal{P}_1(R), -\mathcal{P}_2(R), \Delta(R, T))^T. \tag{2.2}$$

To facilitate notation, we define a partial order $\preceq$ on the domain of f:

$$(R_1, T_1) \preceq (R_2, T_2) \;\Leftrightarrow\; f(R_1, T_1) \leq f(R_2, T_2) \;\Leftrightarrow\; \begin{cases} \mathcal{P}_1(R_1) \geq \mathcal{P}_1(R_2) \\ \mathcal{P}_2(R_1) \geq \mathcal{P}_2(R_2) \\ \Delta(R_1, T_1) \leq \Delta(R_2, T_2). \end{cases}$$

We say that (R_1, T_1) dominates (R_2, T_2) when $(R_1, T_1) \preceq (R_2, T_2)$ and $f(R_1, T_1) \neq f(R_2, T_2)$, i. e., (R_1, T_1) is at least as good as (R_2, T_2) and better with respect to at least one criterion.

If $\mathcal{F} \subseteq \mathcal{R} \times \mathcal{T}$ is a set of combinations of a relation and a transformation, we use $(R,T) \preceq \mathcal{F}$ to denote that $(R,T) \preceq (R',T')$ for all $(R',T') \in \mathcal{F}$.

Note that for an approximation ratio $\alpha \in \mathbb{R}_{\geq 0}$ for the multi-criteria largest common point set objective function, we have $\alpha_3 \in [1,\infty)$ and $\alpha_1, \alpha_2 \in (0,1]$ due to the negativity of f_1, f_2.

Using this notation, we can adapt Algorithm 2.1 to the multi-criteria largest common point set problem, see Algorithm 2.3. As in Algorithm 2.1, all transformations obtained from matching pairs of bases are considered. The approximate Pareto front is continuously updated in the function `updateParetoFront`, which uses the auxiliary variable F. The front $\mathcal{F}$ is augmented by solutions which are not dominated by any other element currently in the approximate Pareto front. Furthermore, dominated elements are removed from it. An implementation is given in Algorithm 2.4 and will be discussed in the following.

Input: Finite point sets $P, Q \subseteq \mathbb{R}^d$
Output: Approximate Pareto front $\mathcal{F} \subseteq \mathcal{R} \times \mathcal{A}$

```
1 F ← ∅
2 F ∈ ℝ^{|P|×|Q|}
3 F(i,j) ← ∞ for all i ∈ {1,...,|P|}, j ∈ {1,...,|Q|}
4 for n = 0 to d do
5 |   for all affinely independent ordered sets B_P ⊆ P with |B_P| = n+1 and all ordered
  |   multisets B_Q ⊆ Q with |B_Q| = n+1 do
6 |   |   T ← T_{B_P,B_Q}
7 |   |   (F, F) ← updateParetoFront(F, F, T)
8 |   end
9 end
```

Algorithm 2.3: approximateParetoFront

Considering the results of the previous section, it is not surprising that this algorithm does indeed deliver an approximation of the Pareto front of the problem with respect to the set of affine transformations $\mathcal{T} := \mathcal{A}$.

Theorem 2.20
For finite point sets $P, Q \subseteq \mathbb{R}^d$, Algorithm 2.3 is a $(1,1,2(d+1))$-approximation algorithm for the Pareto front.

Proof. Let $P, Q \subseteq \mathbb{R}^d$ be finite point sets and $(R,T) \in \mathcal{R} \times \mathcal{A}$ an arbitrary point in the domain of the multi-criteria objective function f. For an ordered wide base B_P of $\mathcal{P}_1(R)$, Theorem 2.13 states that there exists an ordered multiset $B_Q \subseteq \mathcal{P}_2(R) \subseteq Q$ with $|B_Q| = |B_P|$ such that $\Delta\Big(R, T_{B_P,B_Q}\Big) \leq 2(d+1)\Delta(R,T)$. Since the solution $\big(R, T_{B_P,B_Q}\big)$ uses the same relation as (R,T), we have

$$f_i(R,T) = -\mathcal{P}_i(R) = f_i(R, T_{B_{P'},B_{Q'}}) \text{ for } i \in \{1,2\}$$

and

$$f_3(R, T_{B_P,B_Q}) = \Delta\Big(R, T_{B_P,B_Q}\Big) \leq 2(d+1)\Delta(R,T) = 2(d+1) f_3(R,T).$$

If there is no $(R',T') \in \mathcal{F}$ which dominates $\big(R, T_{B_{P'},B_{Q'}}\big)$, the computed solution is adjoined to $\mathcal{F}$ in the function `updateParetoFront`. Consequently, i) of Definition 2.18 is fulfilled and condition ii) is ensured in `updateParetoFront`. □

In the following, we examine the running time of the algorithm. Many multi-criteria optimization problems cannot be solved efficiently simply because the size of the Pareto front is not polynomial in the size of the input. Fortunately, we can give a low upper bound which is not prohibitive for typical applications in our case.

Theorem 2.21
Let $P, Q \subseteq \mathbb{R}^d$ be two finite point sets, $\mathcal{R} \subseteq 2^{P \times Q}$ a set of relations of P and Q and $\mathcal{T}$ a set of admissible transformations of $\mathbb{R}^d$. The size of the Pareto front $\mathcal{F}$ of $\mathcal{R} \times \mathcal{T}$ with respect to the objective function defined in Definition 2.19 is bounded by $|P||Q|$.

Proof. Since the number of possible values for the first two coordinates of the objective function is bounded by $|P|$ and $|Q|$, respectively, there are no more than $|P||Q|$ possible combinations. For each combination of possible sizes of subsets of P and Q, there is a minimal error of a relation covering an appropriate amount of points. □

If we use the set of relations $\mathcal{R} \coloneqq 2^{P \times Q}$, i. e., measure the error between common point sets using the Hausdorff distance, the function `updateParetoFront` can be implemented particularly efficiently, see Algorithm 2.4. We employ the idea used for the proof of Theorem 2.21. For each combination of possible sizes of subsets of P and Q, we store the minimal error of a relation covering an appropriate amount of points in the matrix $F \in \mathbb{R}^{|P| \times |Q|}$. Instead of examining just a single relation R which maximizes $\min\{\mathcal{P}_1(R), \mathcal{P}_2(R)\}$ for a given bound δ and transformation T as in Algorithm 2.1, we investigate maximizing relations R_l for all possible $\delta \geq 0$ in ascending order. This is possible since there are at most $|P| + |Q|$ such relations, cf. Lemma 2.22. Notice that we have $\Delta(R_l, T) = \|T(p_l) - q_l\|$ due to the sorting of R_l. A formulation of the algorithm in pseudo-code is given in Algorithm 2.4.

Position (i, j) of the matrix $F \in \mathbb{R}^{|P| \times |Q|}$ stores the currently best known bound $F(i, j)$ on the Hausdorff distance subject to which there exist common point sets $P' \subseteq P$ and $Q' \subseteq Q$ with $|P'| \geq i$ and $|Q'| \geq j$ and a transformation $T \in \mathcal{T}$ with $d_H(T(P'), Q') \leq F(i, j)$.

With this efficient way to maintain the current approximation of the Pareto front, a multi-criteria approximation can be obtained with only a slight increase in computational effort compared to Algorithm 2.1.

Lemma 2.22
The running time of Algorithm 2.3 is in $\mathcal{O}\Big(d(|P||Q|)^{d+2}(|P| + |Q|)\Big)$ when using Algorithm 2.4 to maintain the current approximation of the Pareto front.

Proof. As in Algorithm 2.1, the main *for loop* in Algorithm 2.3 is iterated $\mathcal{O}\Big(d(|P||Q|)^{d+1}\Big)$ times. We can assume $|P| \geq d$ and $|Q| \geq d$. The computation of the transformation $T \coloneqq T_{B_P, B_Q}$ is in $\mathcal{O}(d^3)$ and dominated by the running time of `updateParetoFront`. The running time of Algorithm 2.4 is determined by evaluating the relations $R_l \subseteq R$ for all $l \in \{1, \ldots, k\}$. Using Lemma 1.7, we can assume $k = |R| \leq |P| + |Q|$. This implies that the running time of `updateParetoFront` is in $\mathcal{O}((|P| + |Q|)|P||Q|)$, which concludes the proof. □

2.5 Adaption to the Application in Mechanical Engineering

In this section, which is partly based on joint work with Andreas Schrottenloher [Sch13], we investigate an approach which is tailored specifically towards our application in mechanical engineering. We modify our approach to consider only a subset of affine transformations, which

Input: $\mathcal{F} \subseteq \mathcal{R}$, $F \in \mathbb{R}^{|P|\times|Q|}$, $T \in \mathcal{A}$
Output: updated front $\mathcal{F} \subseteq \mathcal{R}$, current objective values $F \in \mathbb{R}^{|P|\times|Q|}$

$R \leftarrow \left\{(p,q) \mid p \in P,\ q = \operatorname{argmin}_1 \limits_{q' \in Q} \|T(p) - q'\|\right\}$
$\quad \cup \left\{(p,q) \mid q \in Q,\ p = \operatorname{argmin}_1 \limits_{p' \in P} \|T(p') - q\|\right\}$
Sort R such that $R = \{(p_i, q_i) \mid i = 1, \ldots, k\}$ with $\|T(p_i) - q_i\| \leq \|T(p_{i+1}) - q_{i+1}\|$ for all $i \in \{1, \ldots, k-1\}$
for $l = 1$ *to* k **do**
  $R_l \leftarrow \{(p_i, q_i) \mid i = 1, \ldots, l\}$
  $\delta \leftarrow \|T(p_i) - q_i\|$
  if $F(\mathcal{P}_1(R_l), \mathcal{P}_2(R_l)) \geq \delta$ **then**
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{(R_l, T)\}$
    for $i = 1$ *to* $|\mathcal{P}_1(R_l)|$ **do**
      for $j = 1$ *to* $|\mathcal{P}_2(R_l)|$ **do**
        if $\delta < F(i,j)$ **then**
          $F(i,j) \leftarrow \delta$
        end
      end
    end
  end
end

Algorithm 2.4: updateParetoFront

corresponds to the deformations of sheet metal parts that can be achieved with the methods developed in the scope of the research project, see [Vol+13] for an in-depth treatment.

All sheet metal parts considered in the scope of this project are manufactured by an incremental forming process which forms an initially flat sheet metal part into a desired shape. Due to manufacturing issues, the deformation strategies for this incremental process are currently limited to specific affine transformations. It turns out that the LCP measure can be computed more efficiently when restricted to this class of transformations. At first, we give a formal definition.

Definition 2.23 (Flat transformation)
Let $T\colon \mathbb{R}^n \longrightarrow \mathbb{R}^m$, $T(x) \coloneqq Ax + t$ where $A \in \mathbb{R}^{m\times n}$, $t \in \mathbb{R}^m$, an affine transformation. T is called flat if there exist $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$ such that $\|u\|_2 = 1 = \|v\|_2 = \|Av\|_2$ and

$$\langle Ax, u\rangle = \langle x, v\rangle \qquad \forall x \in \mathbb{R}^n.$$

An example of such a transformation can be seen in Figure 2.5. The flat transformation aligning the car door and the deformed car body is a combination of a rotation and a non-uniform scaling: the door is scaled in vertical direction, rotated and translated to fit the car body. The corresponding vectors u and v are normals to the plane containing the almost flat part of the car door and transformed car door, respectively.

[1]Figure generated from a surface mesh included in the matlab package "Toolbox Graph" by Gabriel Peyré

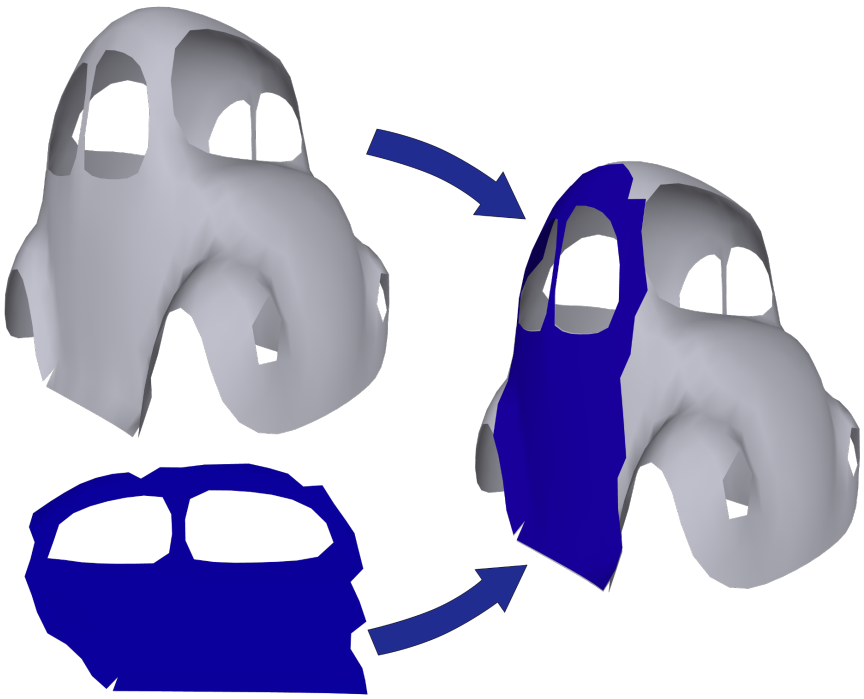


Figure 2.5: Alignment of a car door and body[1] using a flat transformation.

These transformations are equivalently characterized by having a singular value of 1.

Theorem 2.24
Let $T\colon \mathbb{R}^n \longrightarrow \mathbb{R}^m$, $T(x) \coloneqq Ax + t$ where $A \in \mathbb{R}^{m\times n}$, $t \in \mathbb{R}^m$ be an affine transformation. T is flat if and only if A has a singular value of 1.

Proof. Let T be flat and $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$ as above. Since $\|u\|_2 = 1 = \|v\|_2$, there exist orthogonal transformations $U_1 \in O(m)$, $V_1 \in O(n)$ such that $U_1 e_1 = u$ and $V_1 e_1 = v$. Since $1 = \|Av\|_2\|u\|_2 \geq \langle Av, u\rangle = \langle v, v\rangle = 1$, we have $Av = u$. For $D_1 \coloneqq U_1^T A V_1 \in \mathbb{R}^{m\times n}$ and arbitrary $s \in \mathbb{R}^n$ we have

$$\begin{aligned}\langle D_1 s, e_1\rangle &= \left\langle U_1^T A V_1 s, e_1\right\rangle = \langle AV_1 s, U_1 e_1\rangle \\ &= \langle AV_1 s, u\rangle = \langle V_1 s, v\rangle = \langle V_1 s, V_1 e_1\rangle = \langle s, e_1\rangle,\end{aligned}$$

which implies $e_1^T D_1 = e_1^T$. Further, we have

$$D_1 e_1 = U_1^T A v = U_1^T u = e_1,$$

which means that D_1 has the following block structure:

$$D_1 = \left(\begin{array}{c|c} 1 & 0\cdots 0 \\ \hline \begin{matrix}0\\ \vdots \\ 0\end{matrix} & A' \end{array}\right)$$

with $A' \in \mathbb{R}^{(m-1)\times(n-1)}$. Let $A' = U_2 D_2 V_2^T$ be a singular value decomposition of A'. Then

$$U \coloneqq U_1 \left(\begin{array}{c|c} 1 & 0\cdots 0 \\ \hline 0 & \\ \vdots & U_2 \\ 0 & \end{array}\right), \quad D \coloneqq \left(\begin{array}{c|c} 1 & 0\cdots 0 \\ \hline 0 & \\ \vdots & D_2 \\ 0 & \end{array}\right), \quad V \coloneqq V_1 \left(\begin{array}{c|c} 1 & 0\cdots 0 \\ \hline 0 & \\ \vdots & V_2 \\ 0 & \end{array}\right)$$

is a singular value decomposition of A with singular value 1.

For a singular value decomposition $A = UDV^T$ of A with singular value $\sigma_i = 1$, let $v \coloneqq Ve_i$ and $u \coloneqq Ue_i$. We have

$$\langle Ax, u\rangle = \left\langle UDV^T x, u\right\rangle = \left\langle V^T x, D^T U^T u\right\rangle = \left\langle V^T x, De_i\right\rangle = \sigma_i \langle x, Ve_i\rangle = \langle x, v\rangle.$$

The other properties can be easily seen from the definition of u and v. □

Since this class of transformations is a proper subset of the set of affine transformations, we have to adapt Algorithm 2.1 for this new setting. Fortunately, there is a straightforward way to achieve this. Instead of using $d+1$ affinely independent points contained in the set as bases, we consider so-called flat bases.

Definition 2.25 (Flat base)
Let $P \subseteq \mathbb{R}^d$. An ordered set $B_P \coloneqq (p_1, \dots, p_d, p_\perp) \subseteq \mathbb{R}^d$ is called flat base of P if $p_1, \dots, p_d \in P$ are affinely independent and $p_\perp \in \{p_1, \dots, p_d\}^\perp$, $\|p_\perp\|_2 = 1$ is a unit normal of the hyperplane spanned by $p_1, \dots, p_d$.

Note that the unit normal $p_\perp$ is not required to be in P and is characterized by $b_1, \dots, b_d$ up to sign. Thus, we only need to consider ordered sets of d points in P and both their corresponding unit normals in an adaption of Algorithm 2.1, see Algorithm 2.5. The point clouds and their overlapping parts in the engineering application are known to be non-degenerate, so we do not need to consider bases in lower dimensions.

Input: $P, Q \subseteq \mathbb{R}^d$ finite point sets, $\delta \geq 0$
Output: Flat transformation T, relation R

$L \leftarrow -1$
for *all affinely independent ordered sets* $F_P \subseteq P$, $F_Q \subseteq Q$ *with* $|F_P| = d = |F_Q|$ **do**
  for *all unit normals* $p_\perp$ *of* F_P, $q_\perp$ *of* F_Q **do**
    $B_P \leftarrow F_P \cup \{p_\perp\}$, $B_Q \leftarrow F_Q \cup \{q_\perp\}$
    $T' \leftarrow T_{B_P, B_Q}$
    $R' \leftarrow \operatorname{argmax}\{\min\{|\mathcal{P}_1(R')|, |\mathcal{P}_2(R')|\} \mid R' \in \mathcal{R},\ \Delta(R', T') \leq \delta\}$
    if $\min\{|\mathcal{P}_1(R')|, |\mathcal{P}_2(R')|\} > L$ **then**
      $L \leftarrow \min\{|\mathcal{P}_1(R')|, |\mathcal{P}_2(R')|\}$
      $T \leftarrow T'$, $R \leftarrow R'$
    end
  end
end

Algorithm 2.5: flatBaseMatching

Given flat bases B_P, B_Q of point sets $P, Q \subseteq \mathbb{R}^d$, a flat transformation T_{B_P, B_Q} mapping B_P to B_Q can be computed as in Remark 2.5:

Lemma 2.26
Let $B_P \coloneqq (p_1, \dots, p_d, p_\perp)$, $B_Q \coloneqq (q_1, \dots, q_d, q_\perp) \subseteq \mathbb{R}^d$ be flat bases. There exist $A \in \mathbb{R}^{d\times d}$ and $t \in \mathbb{R}^d$ such that

$$Ap_i + t = q_i \quad \forall i \in \{1, \dots, n\} \qquad \wedge \qquad Ap_\perp = q_\perp$$

and the resulting affine transformation is flat.

Proof. The existence of such a transformation is a direct consequence of Remark 2.5. Let $A \in \mathbb{R}^{d\times d}$, $t \in \mathbb{R}^d$ be the accordant linear transformation and translation and consider $x \in \mathbb{R}^d$. Since $p_1, \dots, p_d$ are affinely independent, there exist $\lambda_i \in \mathbb{R}, i \in \{2, \dots, d\}$ and $\lambda_\perp \in \mathbb{R}$ such that

$$x = \sum_{i=2}^{d} \lambda_i (p_i - p_1) + \lambda_\perp p_\perp.$$

Using $A(p_i - p_1) = q_i - q_1$ for $i \in \{2, \dots, d\}$, we have

$$\begin{aligned}
\langle Ax, q_\perp \rangle &= \left\langle A\left(\sum_{i=2}^{d} \lambda_i (p_i - p_1) + \lambda_\perp p_\perp\right), q_\perp \right\rangle \\
&= \left\langle \sum_{i=2}^{d} \lambda_i (q_i - q_1) + \lambda_\perp q_\perp, q_\perp \right\rangle \\
&= \langle \lambda_\perp q_\perp, q_\perp \rangle \\
&= \lambda_\perp \\
&= \langle \lambda_\perp p_\perp, p_\perp \rangle \\
&= \left\langle \sum_{i=2}^{d} \lambda_i (p_i - p_1) + \lambda_\perp p_\perp, p_\perp \right\rangle \\
&= \langle x, p_\perp \rangle.
\end{aligned}$$

With $u \coloneqq q_\perp$ and $v \coloneqq p_\perp$, this is exactly the requirement stated in Definition 2.23. □

Due to the special properties of flat transformations and the non-degeneracy of the parts in our applications, the adapted algorithm terminates after $\mathcal{O}\Big((|P||Q|)^d\Big)$ iterations of the main loop.

Similar to the approach presented in [AMCO08], we can also extend our flat base B to contain an additional point b_0 such that $\{b_0, b_1, \dots, b_d\}$ is affinely dependent. While this increases the number of bases to consider, many combinations of bases can be disregarded since a transformation mapping $d + 2$ affinely dependent points to $d + 2$ affinely dependent points is over-determined:

Aiger, Mitra, and Cohen-Or [AMCO08] show that in dimension $d = 3$, given an affinely dependent set B_P of P with $|B_P| = 4$, it is easy to find all subsets B_Q^i, $i \in \{1, \dots, k\}$ of Q with cardinality 4 which admit an affine transformation mapping B_P to B_Q^i. This is achieved by considering properties of such sets which are invariant with respect to affine transformations. Efficient specialized data structures, so-called range trees, are used to extract suitable sets B_Q^i. Therefore, it is no longer necessary to consider all possible subsets of 4 points of Q as possible bases, but candidates can be efficiently extracted. In practice, this effect outweighs the increased number of possible bases and speeds up the algorithm considerably.

In dimension $d = 3$, this concept can be transferred to flat bases with some modifications to the algorithm. Technical details, implementation and computational results are described in detail in the master's thesis [Sch13], which was co-supervised by the author. The transformation aligning the car door to the car body in Figure 2.5 was computed using this algorithm.

A possible generalization of this method to higher dimensions would be to consider affinely dependent sets $B = \{b_0, \dots, b_d\} \subseteq P$ with $b_1, \dots, b_d$ affinely independent. The point b_0 can then be expressed in barycentric coordinates $\lambda_1, \dots, \lambda_d \in \mathbb{R}^d$, $\sum_{i=1}^{d} \lambda_i = 1$:

$$b_0 = \sum_{i=1}^{d} \lambda_i b_i.$$

Since the barycentric coordinates are invariant with respect to affine transformations, it is sufficient to consider only pairs of bases where the barycentric coordinates of the 0th point are reasonably close. The efficient extraction of bases in both sets which can be matched by an affine transformation is a possible direction for future research.

Chapter 3

Non-Rigid Shape Matching

Encouraged by recent advances in rigid shape matching, a lot of effort has been invested in the more challenging problem of non-rigid shape matching. The most promising approach to this problem is to compare metric spaces which are induced by the manifolds representing the shapes. A suitable measure for the similarity of metric spaces is the Gromov-Hausdorff distance, which has been extensively studied in the field of metric geometry [BBI01]. As mentioned in the introduction, a correspondence between two surfaces yields the amount by which parts of one surface need to be stretched or shrunk to be transformed to the other surface. Considering the importance of the stretching and shrinking operations for the manufacturing process described in Section 1.2, this information could be used as basis for subsequent research investigating more general transformations of sheet metal parts and their manufacturing strategies.

After a short introduction to the problem setting and a few basic definitions, we present several decision versions of the problem. In Section 3.2, we investigate the computational complexity of these decision problems related to the computation of the Gromov-Hausdorff distance. By exploiting connections to other well-known combinatorial optimization problems like the maximum clique and the graph isomorphism problem, we show that the Gromov-Hausdorff distance of two metric spaces is — under standard complexity theoretic assumptions — not approximable within reasonable bounds in polynomial time. Section 3.3 deals with overcoming the complexity issues for several classes of practical instances. The rough structure of most metric spaces occurring in these instances can be described by a small number of points. This can be used — similar in spirit to the method used in Chapter 2 — to compute correspondences between metric spaces from a relation covering just a few points in each set. Papers covering the achieved results are currently in preparation [Sch14b; Sch14a].

3.1 The Gromov-Hausdorff Distance

In traditional shape matching, shapes are usually represented as point clouds in an ambient Euclidean space. An established measure for the similarity of such shapes is the Hausdorff distance under certain transformations, see, e. g., the problem discussed in Chapter 2.

In non-rigid shape matching, however, we are interested in the similarity of shapes as intrinsic properties of surfaces. There is a vast amount of approaches to this topic, including embedding both shapes in an Euclidean space and minimization of various energy functionals, see, e. g., [Tam+13] for a recent survey. Following the approach pioneered by [EK01; Bro+04; MS05], we consider the shapes as metric spaces derived from 2-dimensional manifolds.

The metric is induced by the geodesic distances of points on the shape. Figure 3.1 illustrates that these geodesic distances (and also the geodesics for that matter) are equal or at least close to each other for corresponding points of the person in the two poses.

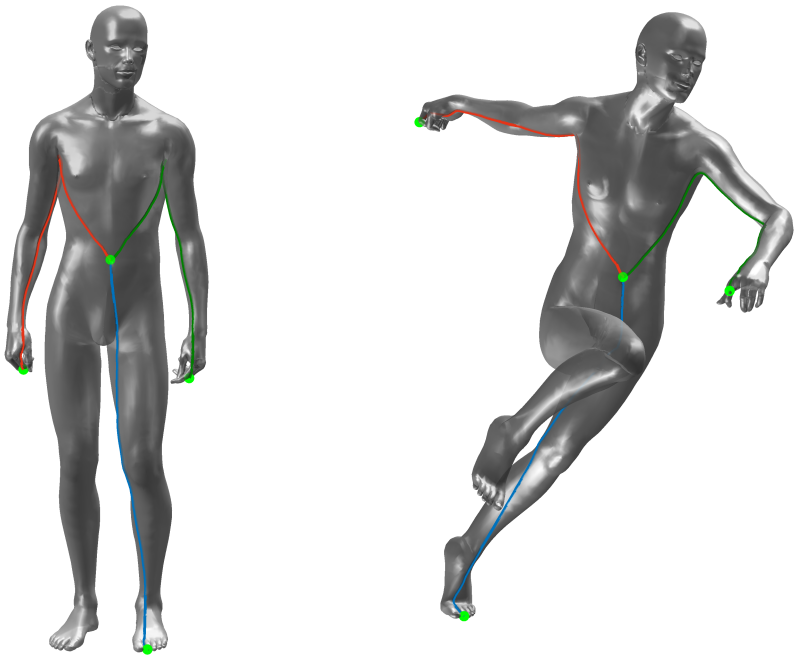


Figure 3.1: Geodesic lines on shapes in different poses. The corresponding geodesic lines of the shapes are very similar and therefore geodesic distances are approximately preserved by the transformation depicted here.

Note that all mesh surfaces used for testing and displaying purposes have been taken from the "Tools for Surface Comparison and Analysis" (TOSCA) project, see [BBK08] for reference.

Shape matching which is also robust with respect to topology changes can be realized by using so-called diffusion distances [Laf04; CL06]. The diffusion distance between two points is defined by properties of the heat diffusion process on manifolds and can be interpreted as an average length of all paths between these points. The distances define a metric on a manifold and have been employed in non-rigid shape matching by Bronstein et al. [Bro+10].

In non-rigid shape matching, we are usually given a shape as a mesh surface. In the following, we derive metric spaces from meshes in the following way:

Definition 3.1
Let M be a mesh. A corresponding finite metric space (X, d_X) is given by (a subset of) the vertices of M and the (geodesic) distance of two points is given by a shortest path on the mesh connecting these points.

Note that an error can be introduced by considering solely the vertices of the mesh, see Lemma 3.8. Furthermore, a mesh is in general not uniquely characterized by its vertices and their geodesic distances, see Figure 3.2.

Several efficient methods for the computation of geodesic distances on such objects in $\mathbb{R}^3$ are known. Exact algorithms for computing the geodesic distances of one source point to all n vertices of a polyhedral surface are given by Mitchell, Mount, and Papadimitriou [MMP87] and Chen and Han [CH96] with running time $\mathcal{O}(n^2 \log(n))$ and $\mathcal{O}(n^2)$, respectively. Surazhsky et al. [Sur+05] present an efficient implementation of the algorithm given in [MMP87], along

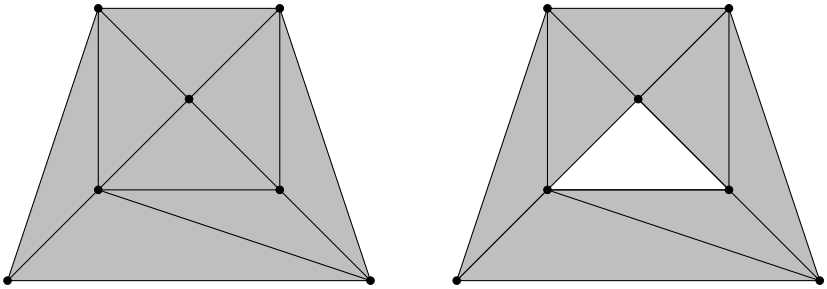


Figure 3.2: Two distinct meshes embedded in $\mathbb{R}^2$ which have identical geodesic distances between corresponding vertices. Faces are depicted in grey, vertices and edges in black.

with a modification to obtain an approximation with bounded error in time $\mathcal{O}(n \log(n))$. Thus, the corresponding metric space can be efficiently derived.

In this chapter, we solely use properties of metric spaces. Therefore, the methods developed here are not limited to metric spaces derived from 2-dimensional surfaces which we use for illustration.

The comparison of metric spaces is an established topic in metric geometry, see [BBI01], where one is mostly interested in asymptotic results concerning the Gromov-Hausdorff distance. The original definition is inspired by the idea of finding a suitable metric space in which both metric spaces can be isometrically embedded. Thus, the Hausdorff distance can be used to measure their similarity.

Definition 3.2
Let $\mathcal{M}_X := (X, d_X)$ and $\mathcal{M}_Y := (Y, d_Y)$ be two compact metric spaces. The Gromov-Hausdorff distance $d_{GH}(\mathcal{M}_X, \mathcal{M}_Y)$ of $\mathcal{M}_X$ and $\mathcal{M}_Y$ is defined as the minimal Hausdorff distance of isometric embeddings of $\mathcal{M}_X$ and $\mathcal{M}_Y$ in a suitable metric space (Z, d_Z), i. e.,

$$d_{GH}(\mathcal{M}_X, \mathcal{M}_Y) := \inf\{d_H(\phi(X), \psi(Y)) \mid (Z, d_Z) \text{ metric space}, \phi \in \mathrm{Iso}(X, Z), \psi \in \mathrm{Iso}(Y, Z)\},$$

where $\mathrm{Iso}(X, Z), \mathrm{Iso}(Y, Z)$ denote the set of isometric embeddings of $\mathcal{M}_X$ and $\mathcal{M}_Y$ in (Z, d_Z), respectively. When the metrics of the spaces can be inferred from the context, we also use the simpler notation

$$d_{GH}(X, Y) := d_{GH}((X, d_X), (Y, d_Y)).$$

While the idea formulated in the above definition is rather descriptive, the formula presented in Definition 3.2 is suited mainly for the topics arising in metric geometry, but hard to treat computationally. Throughout this chapter, we use a reformulation of this definition which generalizes the concept of isometric maps. We use relations and correspondences (cf. Section 1.3) instead of maps and define their *distortion*, which measures how close to an isometry a correspondence is.

In contrast to the measures introduced in Chapter 2, we cannot specify the error introduced by assigning a point $x \in X$ to a point $y \in Y$ directly, since there is no canonical metric space in which these points can be compared.

Therefore, the distortion associated with a relation measures how much distances between pairs of matched points differ:

Definition 3.3 (Distortion)
Let (X, d_X) and (Y, d_Y) be two metric spaces. The distortion $\Gamma(R) \coloneqq \Gamma_\infty(R)$ of a relation $R \subseteq X \times Y$ is defined as the maximal difference between distances of matched points

$$\Gamma(R) = \Gamma_\infty(R) \coloneqq \sup_{\substack{(x,y)\in R \\ (x',y')\in R}} |d_X(x, x') - d_Y(y, y')|.$$

For $p \in \mathbb{N}$ and finite X and Y, the p-distortion of a relation is given by

$$\Gamma_p(R) \coloneqq \sqrt[p]{\frac{1}{(|X| + |Y|)^2} \sum_{\substack{(x,y)\in R \\ (x',y')\in R}} |d_X(x, x') - d_Y(y, y')|^p}.$$

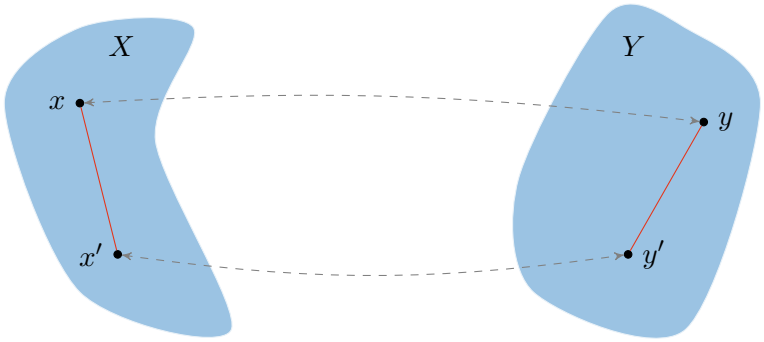


Figure 3.3: Illustration of the distortion induced by the relation $R = \{(x, y), (x', y')\} \subseteq X \times Y$ between X and Y.

The normalizing factor in the definition of the p-distortion is motivated by Lemma 1.7. For the application in mechanical engineering, it could be useful to study a measure which is also invariant with respect to uniform scaling. For a range of scaling factors $[a, b] \subseteq \mathbb{R}_{\geq 0}$, consider the generalized definition

$$\hat{\Gamma}(R) \coloneqq \inf_{\alpha\in[a,b]} \sup_{\substack{(x,y)\in R \\ (x',y')\in R}} |d_X(x, x') - \alpha d_Y(y, y')|.$$

As we will see in the following, finding a correspondence with minimal distortion is already a very difficult problem. Therefore, a direct approach to this problem is not very promising. However, an approximation for the scaling factor can be determined in advance by considering certain properties of both metric spaces, like diameter and local distance sets introduced in Section 3.3.1.

As the Hausdorff distance (see Lemma 1.6), the Gromov-Hausdorff distance can also be expressed using correspondences. For a proof, see [BBI01, Theorem 7.3.25].

Lemma 3.4
Let (X, d_X) and (Y, d_Y) be compact metric spaces, then

$$d_{GH}(X, Y) = \frac{1}{2} \inf_{R\in\mathcal{R}(X,Y)} \Gamma(R) = \frac{1}{2} \inf_{R\in\mathcal{R}(X,Y)} \sup_{\substack{(x,y)\in R \\ (x',y')\in R}} |d_X(x, x') - d_Y(y, y')|,$$

i. e., the Gromov Hausdorff distance is given by one half the infimum of the distortion over all correspondences $R \in \mathcal{R}(X, Y)$ (cf. Definition 1.5) between X and Y.

The factor of $\frac{1}{2}$ is induced by the tight bound $|d_Z(x,x') - d_Z(y,y')| \leq d_Z(x,y) + d_Z(x',y')$ which holds in every metric space (Z, d_Z) and for all $x, x', y, y' \in Z$.

Similar measures for the similarity of metric spaces have also been investigated, see [Mé08].

Definition 3.5
For $p \in \mathbb{N}$ and finite metric spaces (X, d_X), (Y, d_Y), the p-Gromov-Hausdorff distance is defined as

$$d_{GH,p}(X,Y) \coloneqq \frac{1}{2} \min_{R \in \mathcal{R}(X,Y)} \Gamma_p(R)$$

and the bottleneck p-Gromov-Hausdorff distance is given by

$$d_{BGH,p}(X,Y) \coloneqq \frac{1}{2} \min_{R \in \Pi(X,Y)} \Gamma_p(R)$$

if there is a bijection from X to Y, i. e., $|X| = |Y|$, and $d_{BGH,p}(X,Y) \coloneqq \infty$ otherwise. The set of bottleneck correspondences $\Pi(X,Y)$ is defined in Definition 1.5.

As the Gromov-Hausdorff distance, the p-Gromov-Hausdorff distance can be considered as a generalization of the concept of isometry of metric spaces.

Lemma 3.6
Two finite metric spaces (X, d_X), (Y, d_Y) are isometric if and only if $d_{GH,p}(X,Y) = 0$.

Proof. A correspondence R with $\Gamma_p(R) = 0$ must be a bottleneck correspondence, and each bottleneck correspondence induces a bijection and vice versa. □

As implied by the name, the (bottleneck) p-Gromov-Hausdorff distance defines a metric. This is a well-known property of the Gromov-Hausdorff distance [BBI01, Theorem 7.3.30] and can be easily generalized:

Lemma 3.7
The (bottleneck) p-Gromov-Hausdorff distance is a metric on the set of isometry classes of finite metric spaces.

Proof. The p-Gromov-Hausdorff distance is non-negative by definition, and by Lemma 3.6 the p-Gromov-Hausdorff distance of two metric spaces is zero if and only if the spaces are isometric. Symmetry can be easily verified by "reversing" correspondences. Let (X, d_X), (Y, d_Y) and (Z, d_Z) be finite metric spaces and $R_1 \in \mathcal{R}(X,Y), R_2 \in \mathcal{R}(Y,Z)$ arbitrary correspondences. Consider

$$R \coloneqq \{(x,z) \in X \times Z \mid \exists y \in Y : (x,y) \in R_1 \wedge (y,z) \in R_2\}$$

which is easily verified to be a correspondence in $\mathcal{R}(X,Z)$ and satisfies

$$|d_X(x_1,x_2) - d_Z(z_1,z_2)| \leq |d_X(x_1,x_2) - d_Y(y_1,y_2)| + |d_Y(y_1,y_2) - d_Z(z_1,z_2)|$$

for all $(x_1,z_1),(x_2,z_2) \in R$ and any $y_1, y_2 \in Y$; in particular for $(x_1,y_1),(x_2,y_2) \in R_1$ and $(y_1,z_1),(y_2,z_2) \in R_2$. Substituting in the definition and using Minkowski's inequality (triangle inequality for p-norms) yields $\Gamma_p(R) \leq \Gamma_p(R_1) + \Gamma_p(R_2)$. Since R_1 and R_2 were arbitrary, this proves the claim.

The proof for the bottleneck p-Gromov-Hausdorff distance is analogous. □

Since we are considering computational methods, we are restricted to a finite representation of the investigated metric spaces. The metric property of the Gromov-Hausdorff distance implies that the Gromov-Hausdorff distance of discretizations of metric spaces reflects the distance of the original spaces if the discretization is sufficiently dense, see [MS05]. A generalization to the p-Gromov-Hausdorff distance is again straightforward.

Lemma 3.8
Let (X, d_X), (Y, d_Y) be finite metric spaces, $X' \subseteq X$ and $Y' \subseteq Y$ such that

$$d_H(X', X) \leq \varepsilon_X \text{ and } d_H(Y', Y) \leq \varepsilon_Y$$

for $\varepsilon_X, \varepsilon_Y \geq 0$. For the induced metric subspaces $(X', d_X|_{X'})$ and $(Y', d_Y|_{Y'})$, we have

$$d_{GH,p}(X, Y) \leq \varepsilon_X + \varepsilon_Y + d_{GH,p}(X', Y').$$

Proof. First, we show $d_{GH,p}(X, X') \leq \varepsilon_X$. Consider the relation

$$R \coloneqq \left\{ (x', x) \in X' \times X \mid x' = \operatorname*{argmin}_{1,\, x^* \in X'} d_X(x^*, x) \right\},$$

which is a correspondence since $\mathcal{P}_2(R) = X$ by definition and we have $(x', x') \in R$ for all $x' \in X'$ and therefore $\mathcal{P}_1(R) = X'$. Furthermore, the definition of argmin_1 (cf. Section 1.3) ensures $|R| = |X|$.

For $(x'_1, x_1), (x'_2, x_2) \in R$, the construction of R guarantees that $d_X(x'_i, x_i) \leq \varepsilon_X$ for $i \in \{1, 2\}$. We have

$$\begin{aligned} d_X(x'_1, x'_2) - d_X(x_1, x_2) &\leq d_X(x'_1, x_1) + d_X(x_1, x_2) + d_X(x_2, x'_2) - d_X(x_1, x_2) \\ &= d_X(x'_1, x_1) + d_X(x'_2, x_2) \\ &\leq 2\varepsilon_X. \end{aligned}$$

An analogous argument shows $d_X(x_1, x_2) - d_X(x'_1, x'_2) \leq 2\varepsilon_X$, and thus

$$\left| d_X(x'_1, x'_2) - d_X(x_1, x_2) \right| \leq 2\varepsilon_X.$$

In summary, this yields $d_{GH,p}(X, X') \leq \frac{1}{2}\Gamma_p(R) \leq \varepsilon_X$ and $d_{GH,p}(Y, Y') \leq \varepsilon_Y$. This implies

$$\begin{aligned} d_{GH,p}(X, Y) &\leq d_{GH,p}(X, X') + d_{GH,p}(X', Y') \\ &\qquad + d_{GH,p}(Y, Y') \\ &\leq \varepsilon_X + d_{GH,p}(X, Y) + \varepsilon_Y, \end{aligned}$$

which proves the claim. □

Next, we give precise definitions of the problems in question and investigate their computational complexity.

3.2 Computational Complexity

The measure usually considered in the context of non-rigid shape matching is the Gromov-Hausdorff distance, where we seek a correspondence with minimal distortion. For computational reasons, we only consider rational metric spaces (see Definition 1.1) in this section.

Problem 3.9 (Gromov-Hausdorff distance (GHD))
Input: Rational metric spaces (X, d_X), (Y, d_Y), $\delta \in \mathbb{Q}$
Question: Is there a correspondence $R \in \mathcal{R}(X, Y)$ such that $\Gamma(R) \leq \delta/2$?

This problem is obviously a special case (for $p = \infty$) of computing the p-Gromov-Hausdorff distance for fixed $p \in \mathbb{N} \cup \{\infty\}$.

Problem 3.10 (p-Gromov-Hausdorff distance (p-GHD))
Input: Rational metric spaces (X, d_X), (Y, d_Y), $\delta \in \mathbb{Q}$
Question: Is there a correspondence $R \in \mathcal{R}(X, Y)$ such that $\Gamma_p(R) \leq \delta/2$?

Since the above problems proved to be hard to solve in practice, the related problem of computing the bottleneck Gromov-Hausdorff distance has also been investigated [MS05]. For fixed $p \in \mathbb{N} \cup \{\infty\}$, this problem asks for a bottleneck correspondence whose distortion is below a given threshold.

Problem 3.11 (Bottleneck p-Gromov-Hausdorff distance (p-BGHD))
Input: Rational metric spaces (X, d_X), (Y, d_Y), $\delta \in \mathbb{Q}$
Question: Is there a bottleneck correspondence $R \in \Pi(X, Y)$ such that $\Gamma_p(R) \leq \delta/2$?

Another problem of similar form is the subspace Gromov-Hausdorff distance problem, which asks for the maximal induced subspaces of the given metric spaces whose p-Gromov-Hausdorff distance is below a given threshold $\delta \geq 0$ for fixed $p \in \mathbb{N} \cup \{\infty\}$.

Problem 3.12 (Subspace p-Gromov-Hausdorff distance (δ-p-SGHD))
Input: Rational metric spaces (X, d_X), (Y, d_Y), $N \in \mathbb{N}$
Question: Is there a relation $R \subseteq X \times Y$ with $\Gamma_p(R) \leq \delta/2$ and $|R| \geq N$?

Again, we also consider a variant of the problem which requires a bottleneck correspondence between the subspaces.

Problem 3.13 (Subspace bottleneck p-Gromov-Hausdorff distance (δ-p-SBGHD))
Input: Rational metric spaces (X, d_X), (Y, d_Y), $N \in \mathbb{N}$
Question: Is there a bottleneck relation $R \subseteq X \times Y$ with $\Gamma_p(R) \leq \delta/2$ and $|R| \geq N$?

These problems are similar to the maximum subgraph matching problem [GJ79, p. 202], which is known to be $\mathcal{NP}$-hard. They also turn out to be important subproblems for the algorithms developed in Section 3.3 to solve Problem GHD.

It is easy to see that all the above problems are contained in $\mathcal{NP}$, since a non-deterministic Turing machine can "guess" a suitable correspondence or relation for the problem. In order to further evaluate their complexity, we consider the problem of deciding whether two metric spaces are isometric. This can, in fact, be considered as a special version of all problems mentioned above.

Problem 3.14 (Metric space isometry (MI))
Input: Rational metric spaces (X, d_X), (Y, d_Y)
Question: Is there a correspondence $R \in \mathcal{R}(X, Y)$ with $\Gamma(R) = 0$, i. e., are (X, d_X) and (Y, d_Y) isometric?

To compare the difficulty of the above problems, we use the notion of reducibility.

Definition 3.15 (Many-one reduction)
Let P_1, P_2 be two decision problems. Problem P_1 is many-one reducible to P_2 if there exists a polynomial time algorithm A_{P_1,P_2} which, for each instance I_1 of P_1, computes an instance I_2 of P_2 such that I_2 is a YES-instance of P_2 if and only if I_1 is a YES-instance of P_1. In this case, we write $P_1 \leq_m P_2$.

A problem P_1 which is many-one reducible to a problem P_2 is considered to be not significantly more difficult (and probably easier) than P_2.

Definition 3.16 (m-Equivalence)
Two decision problems P_1, P_2 are m-equivalent if they are many-one reducible to one another, i. e., if $P_1 \leq_m P_2$ and $P_2 \leq_m P_1$.

We show (Theorem 3.22) that Problem MI is m-equivalent to the well-known graph isomorphism problem.

Problem 3.17 (Graph isomorphism (GI))
Input: Graphs $G_1 \coloneqq (V_1, E_1)$, $G_2 \coloneqq (V_2, E_2)$
Question: Is there an isomorphism between G_1 and G_2, i. e., $G_1 \cong G_2$?

The graph isomorphism problem is one of the 12 original problems mentioned in [GJ79] neither known to be in $\mathcal{P}$ nor to be $\mathcal{NP}$-complete, despite a considerable amount of work on this open question. It is widely believed that Problem GI lies in-between $\mathcal{P}$ and the set of $\mathcal{NP}$-complete problems. An extensive treatment can be found in [KST93].

It is a common assumption that computing the Gromov-Hausdorff distance of two metric spaces is an $\mathcal{NP}$-hard problem [Mé07]. Bronstein et al. [Bro+10] claim $\mathcal{NP}$-hardness, but we were unable to trace a proof in the literature, despite extensive investigation and contacting the author. The conjectures rely mostly on the observation that Problem GHD has similar structure to other $\mathcal{NP}$-hard optimization problems.

Mémoli [Mé07] observes that the formula for the Gromov-Hausdorff distance of two metric spaces given in Lemma 3.4 is "reminiscent of" the quadratic assignment problem, which is $\mathcal{NP}$-hard [Bur84]. He states that this can be used to "loosely infer something about the inherent complexity" of computing the Gromov-Hausdorff distance.

In this section, we show that the Problems MI and GI are m-equivalent. Given the results on the complexity of Problem GI, this establishes that it is unlikely that there is a polynomial time algorithm for either of these problems; consequently, neither for the more general Problems GHD, p-GHD and p-BGHD, since Problem MI can be easily reduced to any of these.

Moreover, much stronger results can be achieved for the latter problems. We show that the optimization versions of Problem p-GHD and p-BGHD cannot be approximated within any factor exponential in a polynomial of the size of the input, unless Problem GI is in $\mathcal{P}$, cf. Theorem 3.22. Furthermore, Corollary 3.26 gives an inapproximability result — under the assumption that $\mathcal{P} \neq \mathcal{NP}$ — for Problem GHD up to a factor less than 3, even when the input is restricted to metric spaces which are induced by discretizations of plane 2-manifolds. We also show that Problem δ-p-SBGHD is hard to approximate, even if the threshold δ is not considered part of the input, see Theorem 3.28.

First, we establish that the metric space isometry problem is m-equivalent to the graph isomorphism problem.

Lemma 3.18
Problem MI is m-equivalent to Problem GI, i. e., Problem MI can be reduced to Problem GI and vice versa.

Proof. First, we show that GI $\leq_m$ MI. Let $G_1 \coloneqq (V_1, E_1)$ and $G_2 \coloneqq (V_2, E_2)$ be two graphs on $n \in \mathbb{N}$ vertices. We consider the shortest path metrics d_{G_1} and d_{G_2} on G_1 and G_2, see Definition 1.11. The metric spaces (V_1, d_{G_1}) and (V_2, d_{G_2}) are isometric if and only if G_1 and G_2 are isomorphic, as we show in the following.

By definition of d_{G_1}, we have

$$d_{G_1}(v_1, w_1) = 1 \quad \Leftrightarrow \quad \{v_1, w_1\} \in E_1$$

for any $v_1, w_1 \in V_1$ and similarly

$$d_{G_2}(v_2, w_2) = 1 \quad \Leftrightarrow \quad \{v_2, w_2\} \in E_2$$

for any $v_2, w_2 \in V_2$. For an isometry ϕ from (V_1, d_{G_1}) to (V_2, d_{G_2}), this implies

$$\{v_1, w_1\} \in E_1 \Leftrightarrow \{\phi(v_1), \phi(w_1)\} \in E_2,$$

i. e., ϕ is an isomorphism between G_1 and G_2.

If $\phi\colon V_1 \longrightarrow V_2$ is an isomorphism, a shortest path from $v_1 \in V_1$ to $w_1 \in V_1$ in G_1 is mapped to a shortest path in G_2 from $\phi(v_1)$ to $\phi(w_1)$. Therefore, we have $d_{G_1}(v_1, w_1) = d_{G_2}(\phi(v_1), \phi(w_1))$ and this also holds if v_1 and w_1 are in different connected components. This implies that ϕ is an isometry. In summary, we have GI $\leq_m$ MI.

Next, we establish that MI $\leq_m$ GI. This proof is similar to a sketch of a proof for a related problem published in [PS05]. Given two metric spaces (X, d_X) and (Y, d_Y) with $X \coloneqq \{x_1, \ldots, x_n\}$ and $Y \coloneqq \{y_1, \ldots, y_n\}$, we construct graphs (cf. Figure 3.4) by "replacing" distances $d_X(x, x')$ and $d_Y(y, y')$ between two points by a path of length uniquely determined by $d_X(x, x')$ and $d_Y(y, y')$, respectively. Consider the set of distinct distances

$$D \coloneqq \{d_X(x, x') \in \mathbb{Q} \mid x, x' \in X\} \cup \{d_Y(y, y') \in \mathbb{Q} \mid y, y' \in Y\}$$

occurring in the metric spaces. Let $\iota\colon D \longrightarrow \{|D|, \ldots, 2|D| - 1\} \subseteq \mathbb{N}$ be an arbitrary bijective function, which determines the length of a path replacing a distance[1]. For $k, l \in \{1, \ldots, n\}$, $k > l$ and $j \coloneqq \iota(d_X(x_k, x_l))$, we introduce vertices $\left\{x_{kl}^1, \ldots, x_{kl}^j\right\}$ and set

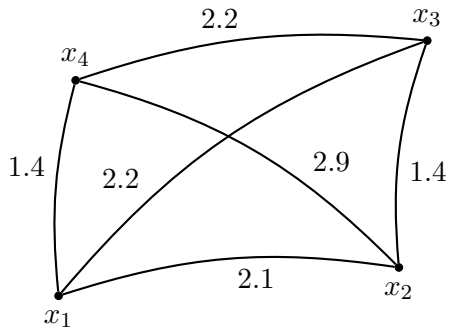
$$V_{kl} \coloneqq \left\{x_{kl}^1, \ldots, x_{kl}^j\right\} \text{ and } E_{kl} \coloneqq \left\{\left\{x_k, x_{kl}^1\right\}, \left\{x_{kl}^1, x_{kl}^2\right\}, \ldots, \left\{x_{kl}^{j-1}, x_{kl}^j\right\}, \left\{x_{kl}^j, x_l\right\}\right\}.$$

Similarly, for $k, l \in \{1, \ldots, n\}$, $k > l$ and $j \coloneqq \iota(d_Y(y_k, y_l))$, we introduce vertices $\left\{y_{kl}^1, \ldots, y_{kl}^j\right\}$. Let

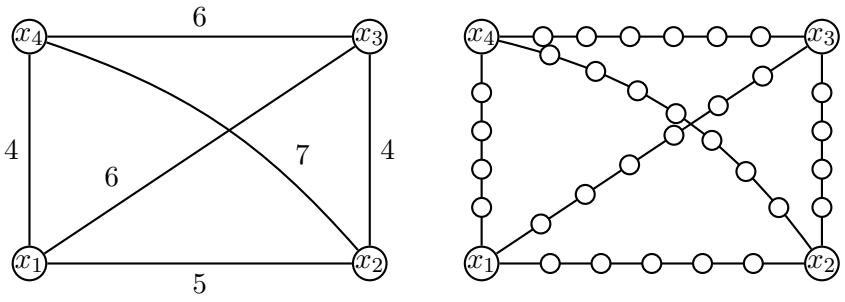
$$W_{kl} \coloneqq \left\{y_{kl}^1, \ldots, y_{kl}^k\right\} \text{ and } F_{kl} \coloneqq \left\{\left\{y_k, y_{kl}^1\right\}, \left\{y_{kl}^1, y_{kl}^2\right\}, \ldots, \left\{y_{kl}^{j-1}, y_{kl}^j\right\}, \left\{y_{kl}^j, y_k\right\}\right\}$$

and

$$V \coloneqq X \cup \bigcup_{k>l} V_{kl}, \quad E \coloneqq \bigcup_{k>l} E_{kl}$$
$$W \coloneqq Y \cup \bigcup_{k>l} W_{kl}, \quad F \coloneqq \bigcup_{k>l} F_{kl}.$$



(a) Input metric space (X, d_X) with distances $L = \{1.4, 2.1, 2.2, 2.9\}$.



(b) Substitution of distances d by labels $\iota(d)$.

(c) Replacement of edges with label $\iota(d)$ by paths with $\iota(d)$ inner vertices.

Figure 3.4: Construction of the graph G_1 from the input metric space (X, d_X) in the proof of Lemma 3.18.

We show that $G_1 \coloneqq (V, E)$ and $G_2 \coloneqq (W, F)$ are isomorphic if and only if (X, d_X) and (Y, d_Y) are isometric.

If G_1 and G_2 are isomorphic, there is an isomorphism $\phi\colon V \longrightarrow W$. Since

$$\iota(d_1) + \iota(d_2) \geq |D| + |D| = 2|D| \geq \iota(d) \text{ for all } d, d_1, d_2 \in D,$$

the triangle inequality holds in G_i, $i \in \{1, 2\}$, and thus $d_{G_1}(x_i, x_j) = \iota(d_X(x_i, x_j))$ for all $x_i, x_j \in X$ and $d_{G_2}(y_i, y_j) = \iota(d_Y(y_i, y_j))$ for all $y_i, y_j \in Y$.

This implies

$$\iota(d_X(x_i, x_j)) = d_{G_1}(x_i, x_j) = d_{G_2}(\phi(x_i), \phi(x_j)) = \iota(d_Y(\phi(x_i), \phi(x_j)))$$

and, since ι is bijective, $\phi|_X$ is an isometry of (X, d_X) and (Y, d_Y).

Suppose (X, d_X) and (Y, d_Y) are isometric and let $\pi\colon \{1, \ldots, n\} \longrightarrow \{1, \ldots, n\}$ be such that $e\colon X \longrightarrow Y$, $e(x_i) \coloneqq y_{\pi(i)}$ is an isometry. Then the map

$$\phi\colon V \longrightarrow W, \quad \phi(v) \coloneqq \begin{cases} y^j_{\pi(k)\pi(l)}, & v = x^j_{kl} \\ y_{\pi(i)}, & v = x_i \end{cases}$$

[1] A construction which uses slightly shorter paths is possible, but omitted here for the sake of simplicity.

is an isomorphism between G_1 and G_2: First of all, it is well-defined since for $x_{kl}^j \in V$ we have $j \leq \iota(d_X(x_k, x_l)) = \iota\Big(d_Y\big(y_{\pi(k)}, y_{\pi(l)}\big)\Big)$ and therefore $y^j_{\pi(k)\pi(l)} \in W$.

Furthermore, for arbitrary $a, b \in V$, we distinguish several cases:

- $a = x_k$ and $b = x_{kl}^1$ for some $k, l \in \{1, \dots, n\}$, $k > l$. We have
$$\{a, b\} = \Big\{x_k, x_{kl}^1\Big\} \in E \text{ and } \{\phi(a), \phi(b)\} = \Big\{y_{\pi(k)}, y^1_{\pi(k)\pi(l)}\Big\} \in F$$
by definition of E_{kl} and $F_{\pi(k)\pi(l)}$.
- $a = x_{kl}^j$ and $b = x_l$ for $k, l \in \{1, \dots, n\}$, $k > l$ and $j = \iota(d_X(x_k, x_l)) = \iota\Big(d_Y\Big(y_{\pi(k)}, y_{\pi(l)}\Big)\Big)$. Again, it holds
$$\{a, b\} = \Big\{x_{kl}^j, x_l\Big\} \in E \text{ and } \{\phi(a), \phi(b)\} = \Big\{y^j_{\pi(k)\pi(l)}, y_{\pi(l)}\Big\} \in F.$$
- $a = x_{kl}^j$ and $b = x_{kl}^{j'}$ for some $k, l \in \{1, \dots, n\}$ $k > l$ and $j, j' \in \mathbb{N}$. We have
$$\begin{aligned}\{a, b\} = \Big\{x_{kl}^j, x_{kl}^{j'}\Big\} \in E &\Leftrightarrow |j - j'| = 1 \\ &\Leftrightarrow \Big\{y^j_{\pi(k)\pi(l)}, y^{j'}_{\pi(k)\pi(l)}\Big\} = \{\phi(a), \phi(b)\} \in F.\end{aligned}$$
- In all other cases, the definitions of E and F imply $\{a, b\} \notin E$ and $\{\phi(a), \phi(b)\} \notin F$.

Therefore, we have $\{a, b\} \in E \Leftrightarrow \{\phi(a), \phi(b)\} \in F$, i. e., $G_1 \cong G_2$. □

Next, we study differences between the Gromov-Hausdorff distance and the bottleneck Gromov-Hausdorff distance, which proves helpful for analysing the computational complexity of these problems. The following lemma states that the bottleneck Gromov-Hausdorff distance is not affected by increasing the distance between all points uniformly, while the Gromov-Hausdorff distance may increase.

Lemma 3.19
Let $\mathcal{M}_X \coloneqq (X, d_X)$, $\mathcal{M}_Y \coloneqq (Y, d_Y)$ be finite metric spaces and $c \geq 0$. Consider the metrics

$$d_X^c(x, x') \coloneqq \begin{cases} d_X(x, x') + c, & x \neq x' \\ 0, & x = x' \end{cases}$$

and

$$d_Y^c(y, y') \coloneqq \begin{cases} d_Y(y, y') + c, & y \neq y' \\ 0, & y = y' \end{cases}$$

on X and Y. Then $\mathcal{M}_X^c \coloneqq (X, d_X^c)$, $\mathcal{M}_Y^c \coloneqq (Y, d_Y^c)$ are metric spaces,

$$d_{GH,p}(\mathcal{M}_X, \mathcal{M}_Y) \leq d_{GH,p}(\mathcal{M}_X^c, \mathcal{M}_Y^c) \leq d_{GH,p}(\mathcal{M}_X, \mathcal{M}_Y) + \frac{c}{2}$$

and

$$d_{BGH,p}(\mathcal{M}_X, \mathcal{M}_Y) = d_{BGH,p}(\mathcal{M}_X^c, \mathcal{M}_Y^c).$$

Proof. It is easy to verify that $\mathcal{M}_X^c$, $\mathcal{M}_Y^c$ are metric spaces. For $x, x' \in X$ and $y, y' \in Y$ it holds

$$|d_X^c(x,x') - d_Y^c(y,y')| = \begin{cases} 0, & x = x' \wedge y = y' \\ |d_X(x,x') - d_Y(y,y')|, & x \neq x' \wedge y \neq y' \\ |d_X(x,x') - d_Y(y,y')| + c, & \text{otherwise,} \end{cases}$$

which proves the result for $p = \infty$. For $p \in \mathbb{N}$, using Lemma 1.7 and the triangle inequality for the p-norm in second to last inequality, we have

$$\begin{aligned} d_{GH,p}(\mathcal{M}_X, \mathcal{M}_Y) &= \frac{1}{2} \min_{R \in \mathcal{R}(X,Y)} \sqrt[p]{\frac{1}{(|X|+|Y|)^2} \sum_{\substack{(x,y)\in R \\ (x',y')\in R}} |d_X(x,x') - d_Y(y,y')|^p} \\ &\leq \frac{1}{2} \min_{R \in \mathcal{R}(X,Y)} \sqrt[p]{\frac{1}{(|X|+|Y|)^2} \sum_{\substack{(x,y)\in R \\ (x',y')\in R}} |d_X^c(x,x') - d_Y^c(y,y')|^p} \\ &= d_{GH,p}(\mathcal{M}_X^c, \mathcal{M}_Y^c) \\ &\leq \frac{1}{2} \min_{R \in \mathcal{R}(X,Y)} \sqrt[p]{\frac{1}{(|X|+|Y|)^2} \sum_{\substack{(x,y)\in R \\ (x',y')\in R}} (|d_X(x,x') - d_Y(y,y')|)^p} + \frac{c}{2} \\ &= d_{GH,p}(\mathcal{M}_X, \mathcal{M}_Y) + \frac{c}{2}. \end{aligned}$$

Since a bottleneck correspondence $R \in \Pi(X,Y)$ satisfies $x \neq x'$ and $y \neq y'$ for all distinct $(x,y), (x',y') \in R$, it holds

$$\begin{aligned} d_{BGH,p}(\mathcal{M}_X, \mathcal{M}_Y) &= \frac{1}{2} \min_{R \in \Pi(X,Y)} \sqrt[p]{\frac{1}{(|X|+|Y|)^2} \sum_{\substack{(x,y)\in R \\ (x',y')\in R}} |d_X(x,x') - d_Y(y,y')|^p} \\ &= \frac{1}{2} \min_{R \in \Pi(X,Y)} \sqrt[p]{\frac{1}{(|X|+|Y|)^2} \sum_{\substack{(x,y)\in R \\ (x',y')\in R}} |d_X^c(x,x') - d_Y^c(y,y')|^p} \\ &= d_{BGH,p}(\mathcal{M}_X^c, \mathcal{M}_Y^c). \end{aligned}$$

□

We use this result to show that the Gromov-Hausdorff distance can be employed to compute the bottleneck Gromov-Hausdorff distance. By uniformly increasing the distances between points, an optimal correspondence can be forced to be bijective, i. e., no single point can be matched to two distinct points.

Lemma 3.20

Problem p-BGHD can be α-approximated given an α-approximation for Problem p-GHD, i. e., an α-approximation of an instance I of Problem p-BGHD can be obtained by computing an α-approximation for a suitable instance of Problem p-GHD which can be derived in time polynomial in the size of I and α.

Proof. Let the metric spaces $\mathcal{M}_X \coloneqq (X, d_X)$, $\mathcal{M}_Y \coloneqq (Y, d_Y)$ be an instance of Problem p-BGHD for $p \in \mathbb{N}$. We can assume $|X| = |Y|$ without loss of generality. Consider the metric spaces $\mathcal{M}_X^c \coloneqq (X, d_X^c)$, $\mathcal{M}_Y^c \coloneqq (Y, d_Y^c)$ defined in Lemma 3.19 for $c \in \mathbb{N}$ with

$$c > 2\sqrt[p]{(|X| + |Y|)^2} \cdot d_{BGH,p}(\mathcal{M}_X, \mathcal{M}_Y),$$

e. g., $c \coloneqq 2\left\lceil \sqrt[p]{(|X| + |Y|)^2} \cdot (\operatorname{diam}(\mathcal{M}_X) + \operatorname{diam}(\mathcal{M}_Y) + 1) \right\rceil$. A correspondence $R \in \mathcal{R}(X, Y)$ which is optimal for $\mathcal{M}_X^c$ and $\mathcal{M}_Y^c$ is a bottleneck correspondence: Suppose $(x, y), (x, y') \in R$ with $y \neq y'$, then

$$\begin{aligned}\Gamma_p(\{(x, y), (x, y')\}) &= \sqrt[p]{\frac{2}{(|X| + |Y|)^2}} \cdot |d_X^c(x, x) - d_Y^c(y, y')| \geq \sqrt[p]{\frac{1}{(|X| + |Y|)^2}} \cdot c \\ &> 2 \cdot d_{BGH,p}(\mathcal{M}_X, \mathcal{M}_Y) = 2 \cdot d_{BGH,p}(\mathcal{M}_X^c, \mathcal{M}_Y^c) \geq 2 \cdot d_{GH,p}(\mathcal{M}_X^c, \mathcal{M}_Y^c),\end{aligned}$$

which is a contradiction. An analogous argument shows that for $(x, y), (x', y) \in R$ we have $x = x'$. Furthermore, using $c > 2\alpha\sqrt[p]{(|X| + |Y|)^2} \cdot d_{BGH,p}(\mathcal{M}_X, \mathcal{M}_Y)$, the above argument also holds for any α-approximation of the optimal solution, which concludes the proof. In case $p = \infty$, the argument can be easily adapted. □

On the other hand, the bottleneck Gromov-Hausdorff distance is obviously an upper bound for the Gromov-Hausdorff distance, but in general no further information can be gained.

Remark 3.21
The bottleneck Gromov-Hausdorff distance does not admit a constant factor approximation for the Gromov-Hausdorff distance, even for subsets of the reals: For $\varepsilon \in (0, 1)$, let

$$X \coloneqq \{-\varepsilon, \varepsilon, 1\} \subseteq \mathbb{R} \text{ and } Y \coloneqq \{0, 1 - \varepsilon, 1 + \varepsilon\} \subseteq \mathbb{R}.$$

We have (cf. Figure 3.5)

$$d_{GH,p}(X, Y) = \frac{1}{2}\Gamma_p(R_{GH}) = \frac{1}{2}\sqrt[p]{\frac{8}{36}|2\varepsilon|^p} \leq \varepsilon$$

and

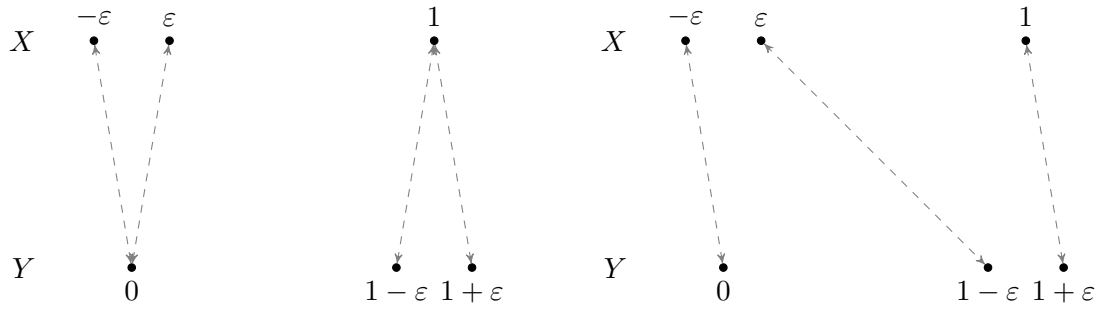
$$d_{BGH,p}(X, Y) = \frac{1}{2}\Gamma_p(R_{BGH}) = \frac{1}{2}\sqrt[p]{\frac{4}{36}|1 - 3\varepsilon|^p} \geq \frac{1}{18}(1 - 3\varepsilon),$$

i. e., the ratio $\frac{d_{BGH,p}(X,Y)}{d_{GH,p}(X,Y)}$ grows arbitrarily large for $\varepsilon \to 0$.

Next, we show that approximating the bottleneck Gromov-Hausdorff distance is at least as hard as solving the graph isomorphism problem.

Theorem 3.22
For any $p \in \mathbb{N} \cup \{\infty\}$, Problem p-BGHD cannot be approximated in polynomial time within any factor $\alpha \in \mathcal{O}\left(2^{\operatorname{poly}(\langle I \rangle)}\right)$, where $\langle I \rangle$ denotes the size of the instance of Problem p-BGHD, unless Problem GI is in $\mathcal{P}$.

Proof. We prove the result for $p \in \mathbb{N}$. The proof for $p = \infty$ can be obtained by a straightforward modification. Assume there is $\alpha \in \mathbb{N}$ such that there exists a polynomial time α-approximation algorithm for Problem p-BGHD. Let $G_i \coloneqq (V_i, E_i)$, $i \in \{0, 1\}$ be two graphs



(a) Optimal correspondence $R_{GH} \in \mathcal{R}(X, Y)$.

(b) Optimal bottleneck correspondence $R_{BGH} \in \Pi(X, Y)$.

Figure 3.5: The Gromov-Hausdorff distance of X and Y can be made arbitrarily small for $\varepsilon \to 0$ while their bottleneck Gromov-Hausdorff distance is bounded from below for small $\varepsilon > 0$. The correspondences inducing these distances are depicted as dashed lines.

on n vertices. For $\beta \in \mathbb{N}$, $\beta > \alpha \cdot \sqrt[p]{(|X| + |Y|)^2} + 1$, consider the metric spaces $\mathcal{M}_i \coloneqq (V_i, d_i)$ with metrics defined by

$$d_i(v_k, v_l) \coloneqq \beta \cdot d_{G_i}(v_k, v_l) + i \text{ for } v_k \neq v_l,$$

and $d_i(v_k, v_l) \coloneqq 0$ otherwise. An isomorphism $\phi : V_0 \longrightarrow V_1$ between G_0 and G_1 induces a bottleneck correspondence $R \coloneqq \{(v, \phi(v)) \mid v \in V_0\}$ with distortion

$$\Gamma_p(R) = \sqrt[p]{\frac{1}{(|V_0| + |V_1|)^2} \sum_{v_0, v_0' \in V_0} \underbrace{|d_0(v_0, v_0') - d_1(\phi(v_0), \phi(v_0'))|^p}_{\leq 1}} \leq 1.$$

This implies $d_{BGH,p}(\mathcal{M}_0, \mathcal{M}_1) \leq 1/2$.

Assume G_0 and G_1 are not isomorphic and let $R \in \Pi(V_0, V_1)$ be a bottleneck correspondence. Since $G_0 \ncong G_1$, there exist $v_0, w_0 \in V_0$ with $(v_0, v_1), (w_0, w_1) \in R$ for some $v_1, w_1 \in V_1$, such that either

$$\{v_0, w_0\} \in E_0 \wedge \{v_1, w_1\} \notin E_1$$

or

$$\{v_0, w_0\} \notin E_0 \wedge \{v_1, w_1\} \in E_1,$$

implying

$$\Gamma_p(R) \geq \frac{|d_0(v_0, w_0) - d_1(v_1, w_1)|}{\sqrt[p]{(|X| + |Y|)^2}} \geq \frac{|\beta \cdot 2 - (\beta \cdot 1 + 1)|}{\sqrt[p]{(|X| + |Y|)^2}} = \frac{\beta - 1}{\sqrt[p]{(|X| + |Y|)^2}} > \alpha,$$

i. e., $d_{BGH,p}(\mathcal{M}_0, \mathcal{M}_1) > 1/2 \cdot \alpha$.

An α-approximation algorithm for Problem p-BGHD could thus be used to decide Problem GI. The construction is polynomial in $\langle I \rangle$ for $\alpha \in \mathcal{O}\left(2^{\text{poly}(\langle I \rangle)}\right)$ since $\langle \alpha \rangle = \lceil \ln(\alpha) \rceil$. □

Using Lemma 3.20, a similar result for Problem p-GHD is an immediate consequence.

Corollary 3.23
Problem p-GHD cannot be approximated within any factor $\alpha \in \mathcal{O}\left(2^{\text{poly}(\langle I \rangle)}\right)$ in polynomial time, unless Problem GI is in $\mathcal{P}$.

Next, we show an inapproximability result for an approximation factor less than 3 under the weaker assumption $\mathcal{P} \neq \mathcal{NP}$.

Theorem 3.24
For $p = \infty$, Problem p-BGHD cannot be approximated within any factor less than 3 in polynomial time, unless $\mathcal{P} = \mathcal{NP}$. This result holds even if the input metric spaces are restricted to discretizations of 2-manifolds embedded in $\mathbb{R}^2$.

Proof. We prove the theorem by reduction of the well-known 3-partition problem.

Problem 3.25 (3-partition)
Input: $N, B \in \mathbb{N}, S = \{s_1, \ldots, s_{3N}\} \subseteq \mathbb{N}$ a multiset, such that ${}^{B}/_{4} < s_i < {}^{B}/_{2}$ and $\sum_{s \in S} s = NB$.
Question: Is there a partition $S_1 \dot\cup \ldots \dot\cup S_N = S$ of S such that $|S_i| = 3$ and $\sum_{s \in S_i} s = B$ for all $i \in \{1, \ldots, N\}$?

Since this problem is strongly $\mathcal{NP}$-complete, we can assume that $B \in \mathbb{N}$ (and not just the encoding length $\langle B \rangle = \lceil \ln(B) \rceil$) is bounded by a polynomial in the size of the input, cf. [GJ79, pp. 96–100], where an explicit bound is given.

For an instance $I \coloneqq (N, B, s_1, \ldots, s_{3N})$ of 3-partition, we construct an instance of Problem p-BGHD as follows:

For each $i \in \{1, \ldots, N\}$, we introduce B distinct points $x_{i1}, \ldots, x_{iB}$ to encode the multiset S_i. Similarly, s_k distinct points $y_{k1}, \ldots, y_{ks_k}$ encode the number s_k for each $k \in \{1, \ldots, 3N\}$. The metric spaces are then defined by (X, d_X) where $X \coloneqq \{x_{ij} \mid i \in \{1, \ldots, N\}, j \in \{1, \ldots, B\}\}$,

$$d_X(x_{ij}, x_{i'j'}) \coloneqq \begin{cases} 0, & i = i' \ \wedge \ j = j' \\ \delta_X, & i = i' \ \wedge \ j \neq j' \\ \Delta_X, & i \neq i' \end{cases}$$

and (Y, d_Y) with $Y \coloneqq \{y_{kl} \mid k \in \{1, \ldots, 3N\}, l \in \{1, \ldots, s_k\}\}$,

$$d_Y(y_{kl}, y_{k'l'}) \coloneqq \begin{cases} 0, & k = k' \ \wedge \ l = l' \\ \delta_Y, & k = k' \ \wedge \ l \neq l' \\ \Delta_Y, & k \neq k', \end{cases}$$

where $\Delta_X > \delta_X > 0$ and $\Delta_Y > \delta_Y > 0$ are constants to be determined later on. An illustration of the spaces can be seen in Figure 3.7.

A bottleneck correspondence $R \in \Pi(X, Y)$ can then be interpreted as an assignment of numbers s_k to multisets S_i, as we will see in the following.

We prove that

$$d_{BGH,\infty}(X, Y) \leq \frac{1}{2} \max\{|\Delta_X - \Delta_Y|, |\Delta_Y - \delta_X|, |\delta_X - \delta_Y|\}$$

if there exists a 3-partition of S. Otherwise, i. e., if the instance I does not admit a 3-partition, we show that $d_{BGH,\infty}(X, Y) \geq \frac{1}{2}|\Delta_X - \delta_Y|$. Therefore, a polynomial time approximation algorithm for Problem p-BGHD with approximation factor less than

$$\frac{|\Delta_X - \delta_Y|}{\max\{|\Delta_X - \Delta_Y|, |\Delta_Y - \delta_X|, |\delta_X - \delta_Y|\}}$$

would imply a polynomial time algorithm for 3-partition. This yields a factor of 3 for $\Delta_X \coloneqq 4, \Delta_Y \coloneqq 3, \delta_X \coloneqq 2, \delta_Y \coloneqq 1$.

Let $S_1 \dot\cup \ldots \dot\cup S_N = S$ be a 3-partition of S and $\sigma_{ik} \coloneqq \sum_{\substack{k' \in \{1,\ldots,k-1\} \\ s_{k'} \in S_i}} s_{k'}$. Consider the relation

$$R \coloneqq \{(x_{ij}, y_{kl}) \in X \times Y \mid s_k \in S_i \ \wedge \ j = \sigma_{ik} + l\}.$$

The points $y_{k1}, \ldots, y_{ks_k}$ representing the number $s_k \in S_i$ are assigned to points $x_{ij}, \ldots, x_{ij'}$, where j and j' are chosen such that $s_{k_1}, s_{k_2}, s_{k_3} \in S_i$ are assigned consecutively. An example is given in Figure 3.6.

In fact, R is a bottleneck correspondence: for $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, B\}$, there is a unique $k \in \{1, \ldots, N\}$ such that $s_k \in S_i$ and $l \coloneqq j - \sigma_{ik} \leq s_k$.

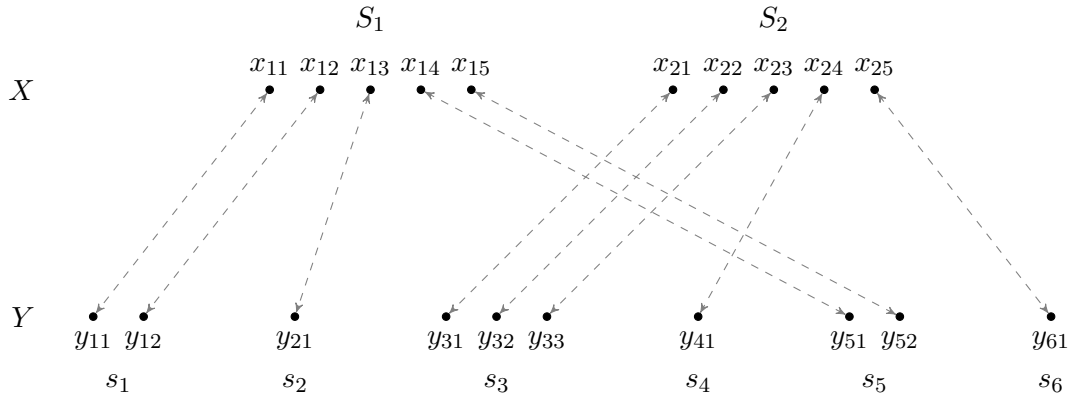


Figure 3.6: Depiction of the constructed metric spaces (X, d_X), (Y, d_Y) (in black) for the 3-partition instance $S \coloneqq \{s_1, \ldots, s_6\} \coloneqq \{2, 1, 3, 1, 2, 1\}$, $B \coloneqq 5$ and the bottleneck correspondence R (grey) associated with the valid 3-partition $S_1 \coloneqq \{s_1, s_2, s_5\}$, $S_2 \coloneqq \{s_3, s_4, s_6\}$.

Let $(x_{ij}, y_{kl}), (x_{i'j'}, y_{k'l'}) \in R$ be two distinct elements, i. e., $x_{ij} \neq x_{i'j'}$ and $y_{kl} \neq y_{k'l'}$. We consider two cases: For $i \neq i'$, we have $d_X(x_{ij}, x_{i'j'}) = \Delta_X$. Assume $k = k'$, then, by definition of R, we have $s_k = s_{k'} \in S_i \cap S_{i'}$. This is a contradiction to $S_1 \dot\cup \ldots \dot\cup S_N = S$ being a 3-partition of S.

For $k \neq k'$, $d_Y(y_{kl}, y_{k'l'}) = \Delta_Y$ holds by definition and therefore

$$\left|d_X(x_{ij}, x_{i'j'}) - d_Y(y_{kl}, y_{k'l'})\right| = |\Delta_X - \Delta_Y|.$$

In case $i = i'$, we have $d_X(x_{ij}, x_{i'j'}) = \delta_X$. This implies

$$\left|d_X(x_{ij}, x_{i'j'}) - d_Y(y_{kl}, y_{k'l'})\right| \in \{|\Delta_Y - \delta_X|, |\delta_X - \delta_Y|\}.$$

Overall, we have $d_{BGH,\infty}(X, Y) \leq \frac{1}{2}\Gamma(R) \in \left\{\frac{1}{2}|\Delta_X - \Delta_Y|, \frac{1}{2}|\Delta_Y - \delta_X|, \frac{1}{2}|\delta_X - \delta_Y|\right\}$.

On the other hand, for a bottleneck correspondence $R \in \Pi(X, Y)$ with $\Gamma(R) < |\Delta_X - \delta_Y|$, let

$$S_i \coloneqq \{s_k \in S \mid \exists l \in \{1, \ldots, s_k\}, \exists j \in \{1, \ldots, B\} \text{ such that } (x_{ij}, y_{kl}) \in R\}.$$

First, we prove that

$$S_i \overset{*}{=} \{s_k \in S \mid \forall l \in \{1, \ldots, s_k\}, \exists_1 j \in \{1, \ldots, B\} \text{ such that } (x_{ij}, y_{kl}) \in R\}.$$

Let $s_k \in S_i$, i. e., $(x_{ij}, y_{kl}) \in R$ for some j, l. As R is a bottleneck correspondence, for each l' there are unique i', j' such that $(x_{i'j'}, y_{kl'}) \in R$. We have

$$|d_X(x_{ij}, x_{i'j'}) - d_Y(y_{kl}, y_{kl'})| = |d_X(x_{ij}, x_{i'j'}) - \delta_Y| \leq \Gamma(R) < |\Delta_X - \delta_Y|$$

and therefore $i = i'$, i. e., all points $y_{k1}, \ldots, y_{ks_k}$ representing the number s_k are assigned to the same set S_i.

This also proves that $S_1, \ldots, S_N$ indeed defines a feasible 3-partition of S: Let $s_k \in S_i \cap S_{i'}$; by $(*)$ we have for all $l \in \{1, \ldots, s_k\}$

$$\exists j, j' \in \{1, \ldots, B\} \text{ such that } (x_{ij}, y_{kl}), (x_{i'j'}, y_{kl}) \in R.$$

Since R is a bottleneck correspondence, we have $i = i'$.

In addition, for $S_i = \{s_{k_1}, \ldots, s_{k_n}\}$ and using $(*)$ for the second equality, we have

$$\begin{aligned}
\sum_{s \in S_i} s &= \sum_{m=1}^{n} s_{k_m} \\
&= \sum_{m=1}^{n} \left| \{(x_{ij}, y_{k_m l}) \mid j \in \{1, \ldots, B\}, l \in \{1, \ldots, s_{k_m}\}\} \right| \\
&= \left| \{(x_{ij}, y_{k_m l}) \mid j \in \{1, \ldots, B\}, m \in \{1, \ldots, n\}, l \in \{1, \ldots, s_{k_m}\}\} \right| \\
&= \left| \{x_{ij} \mid j \in \{1, \ldots, B\}\} \right| = B,
\end{aligned}$$

i. e., the constraint on the sums of the elements of the partitions is fulfilled. The upper and lower bounds on each number s_i imply that each multiset S_i contains exactly 3 elements.

To see that this result holds also for discretizations of 2-manifolds embedded in $\mathbb{R}^2$, first note that (X, d_X) and (Y, d_Y) can be considered as discretizations of metric spaces (X', d_X), (Y', d_Y) consisting of line segments in $\mathbb{R}^2$, see Figure 3.7, and the distances between points are given by the length of a shortest path connecting them. The metric spaces (X', d_X), (Y', d_Y) are in turn subsets of the Euclidean plane.

This also proves the result for discretizations of plane 2-manifolds:

$$X'' \coloneqq X' + \varepsilon \mathbb{B}_2 = \left\{ x \in \mathbb{R}^2 \mid \exists x' \in X' \text{ with } \|x - x'\| \leq \varepsilon \right\}$$

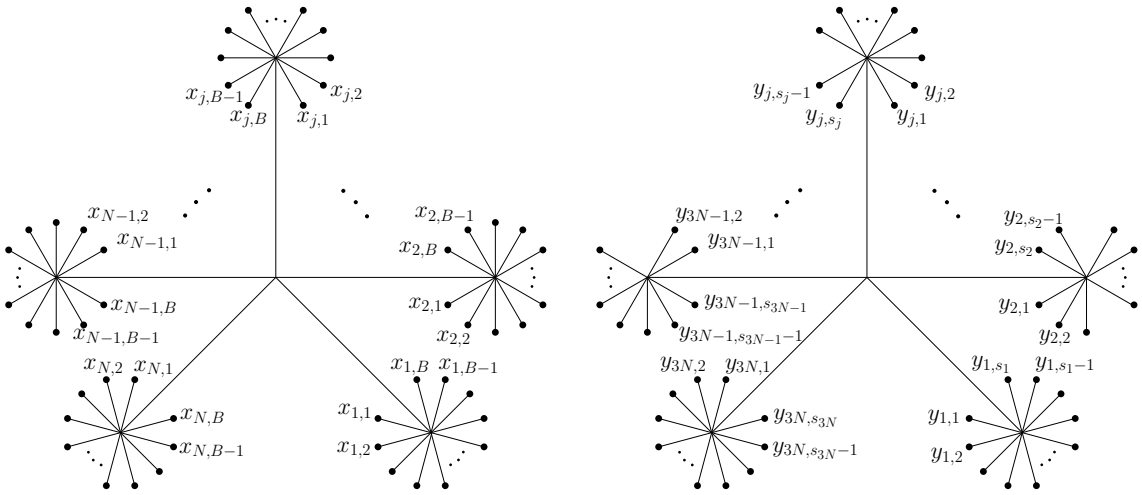
and

$$Y'' \coloneqq Y' + \varepsilon \mathbb{B}_2 = \left\{ y \in \mathbb{R}^2 \mid \exists y' \in y' \text{ with } \|y - y'\| \leq \varepsilon \right\}$$

are smooth 2-manifolds embedded in $\mathbb{R}^2$. The difference between distances of points in (X, d_X), (Y, d_Y) and geodesic distances of corresponding points in X'', Y'' and can be made arbitrarily small by means of ε. Furthermore, we have $X \subseteq X' \subseteq X''$ and $Y \subseteq Y' \subseteq Y''$, i. e., X and Y can be considered as discretizations of 2-manifolds embedded in $\mathbb{R}^2$. □

Again, Lemma 3.20 implies that an analogous result holds for Problem GHD.

Corollary 3.26
Problem GHD cannot be approximated within any factor less than 3 *in polynomial time, unless* $\mathcal{P} = \mathcal{NP}$.



(a) Realization of X **(b)** Realization of Y

Figure 3.7: Encoding of 3-partition in metric spaces.

In the literature, another definition of distortion and a corresponding matching problem are also considered.

Remark 3.27
For an alternative definition of distortion for bottleneck correspondences given by

$$\operatorname{dist}(R) \coloneqq \max_{\substack{(x,y)\in R\\(x',y')\in R}} \max\left\{\frac{d_X(x,x')}{d_Y(y,y')}, \frac{d_Y(y,y')}{d_X(x,x')}\right\},$$

Papadimitriou and Safra [PS05] show — by a reduction of graph 3-colourability — that it is $\mathcal{NP}$-hard to approximate the minimal distortion of a bottleneck correspondence $R \in \Pi(X,Y)$ between finite point sets $X, Y \subseteq \mathbb{R}^3$ with Euclidean distances up to a factor less than 3.

Modifying the parameters in the proof from above, we can achieve an even stronger inapproximability result for plane 2-manifolds (instead of subspaces of $\mathbb{R}^3$): Using the definitions from the proof, we have $\operatorname{dist}(R) = \frac{\Delta_X}{\delta_Y}$ if there is no valid 3-partition, and $\operatorname{dist}(R) \leq \max\left\{\frac{\Delta_X}{\Delta_Y}, \frac{\Delta_Y}{\delta_X}, \frac{\delta_X}{\delta_Y}\right\}$ otherwise. Parameters $\Delta_X \coloneqq \alpha \cdot \Delta_Y, \Delta_Y \coloneqq \alpha \cdot \delta_X, \delta_X \coloneqq \alpha \cdot \delta_Y$ yield that an approximation ratio of $\alpha^2 = \frac{\alpha^3}{\alpha}$ is sufficient to decide 3-partition for an arbitrary $\alpha \in \mathcal{O}\left(2^{\operatorname{poly}(\langle I\rangle)}\right)$, where $\langle I\rangle$ denotes the size of an instance I.

Not surprisingly, the problem of finding maximal subspaces of two metric spaces subject to a bound on their (bottleneck) Gromov-Hausdorff distance is also hard.

Theorem 3.28
For any $\delta \geq 0$ and $p \in \mathbb{N} \cup \{\infty\}$, Problem δ-p-SGHD and Problem δ-p-SBGHD cannot be approximated in polynomial time within any factor less than $(\max\{|X|,|Y|\})^{1/2-\varepsilon}$ for arbitrary $\varepsilon > 0$, unless $\mathcal{P} = \mathcal{NP}$.

Proof. We prove the result for Problem δ-p-SGHD and $p \in \mathbb{N}$ by reduction of the maximum clique problem. The maximum clique problem is known to be hard to approximate within any factor less than $|V|^{1/2-\varepsilon}$, unless $\mathcal{P} = \mathcal{NP}$, see [Hå99].

Problem 3.29 (Maximum clique (MCQ))
Input: Graph $G \coloneqq (V, E)$
Question: What is the biggest $N \in \mathbb{N}$ such that G contains a clique C of size $|C| \geq N$?

Let $G \coloneqq (V, E)$ be an instance of maximum clique. Let $\beta \in \mathbb{N}$, $\beta > \delta \cdot \sqrt[p]{(|X| + |Y|)^2}$,

$$\mathcal{M}_X \coloneqq (V, \beta \cdot d_G) \text{ and } \mathcal{M}_Y \coloneqq (V, \beta \cdot d_D),$$

where d_D denotes the discrete metric

$$d_D(v, w) \coloneqq \begin{cases} 1, & v \neq w \\ 0, & v = w \end{cases}$$

on V and d_G again denotes the shortest path distance on G, cf. Definition 1.11.

We establish that $C \subseteq V$ is a clique of G if and only if there is a relation $R \subseteq X \times Y$ which satisfies $\Gamma_p(R) \leq \delta$.

First, let $C \subseteq V$ be a clique of G and consider the (bottleneck) relation

$$R \coloneqq \{(x, x) \in V \times V \mid x \in C\}.$$

It holds

$$\left| \beta \cdot \underbrace{d_G(x, x')}_{=1} - \beta \cdot \underbrace{d_D(x, x')}_{=1} \right| = 0$$

for all $x, x' \in C$ and therefore $\Gamma_p(R) = 0$. The size of the relation R is obviously $|C|$.

On the other hand, if R is a relation with $\Gamma_p(R) \leq \delta$, we show that

$$C \coloneqq \mathcal{P}_1(R) = \{x \in V \mid \exists y \in V \text{ such that } (x, y) \in R\}$$

is a clique of size $|C| = |R|$ in G. First of all, R is a bottleneck relation: Let $(x, y), (x, y') \in R$ and suppose $y \neq y'$. We have

$$\Gamma_p(R) \geq \sqrt[p]{\frac{1}{(|X| + |Y|)^2}} \cdot \left| \beta \cdot \underbrace{d_G(x, x)}_{=0} - \beta \cdot \underbrace{d_D(y, y')}_{=1} \right| = \frac{\beta}{\sqrt[p]{(|X| + |Y|)^2}} > \delta,$$

which is a contradiction. The same argument can be used to show that $(x, y), (x', y) \in R$ implies $x = x'$. Thus R is a bottleneck relation and therefore $|C| = |R|$.

Next, assume that $x, x' \in C$, $x \neq x'$ are not adjacent, i. e., $\{x, x'\} \notin E$. This implies $d_G(x, x') \geq 2$, and with $d_D(y, y') \leq 1$ for all $y, y' \in V$ we have

$$\Gamma_p(R) \geq \sqrt[p]{\frac{1}{(|X| + |Y|)^2}} \left| \beta \cdot d_G(x, x') - \beta \cdot d_D(y, y') \right| > \delta \quad \forall y, y' \in V,$$

which is again a contradiction.

Therefore we can use an approximation algorithm for Problem δ-p-SGHD to approximate Problem MCQ with the same approximation factor. The proof also shows the result for Problem δ-p-SBGHD and / or $p = \infty$ (using $\beta > \delta$). □

Despite these rather negative results, many metric spaces arising in practical applications have rather nice structure. The basic structure of these spaces is defined by small subsets of points. This decreases the complexity of these problems in practice dramatically. The next section describes these attributes and presents algorithms for Problem GHD which exploit these properties.

3.3 Algorithmic Approaches

Much effort has been invested in methods for the computation of the Gromov-Hausdorff distance as a measure for the similarity of shapes in computer vision. Bronstein, Bronstein, and Kimmel [BBK06] and Elad and Kimmel [EK01] use a generalized multi-dimensional scaling approach, which tries to embed both metric spaces in a common 3-dimensional Euclidean space while preserving the distance matrices as well as possible. The resulting point clouds can then be compared using measures for similarity in $\mathbb{R}^3$, see Chapter 2. They formulate a corresponding nonlinear optimization problem and present a heuristic algorithm for this particular task.

Mémoli and Sapiro [MS05] give conditions subject to which the bottleneck Gromov-Hausdorff distance can be used to bound the Gromov-Hausdorff distance. Further, they present a heuristic algorithm based on these considerations.

In [Mé07], the approximation of the Gromov-Hausdorff distance by a mass transportation problem is studied. This leads to a generally non-convex quadratic optimization problem which is heuristically solved by local optimization techniques.

Raviv, Dubrovina, and Kimmel [RDK12] formulate a slightly more general problem, incorporating a feature vector for each point, as quadratic integer programming problem. They mention that the problem is computationally "almost infeasible" even for small instances of at most 30 vertices. Therefore, they suggest relaxing the integrality constraints and rounding. For a similar problem, Wang et al. [Wan+12] use a formulation introduced in [TKR08] and model the problem as a graph labelling problem. This problem is heuristically solved by employing a dual decomposition approach.

An approach where one tries to minimize the distortion of the area of surface patches induced by a correspondence instead of the distortion of distances between points is considered by Windheuser et al. [Win+11]. This results in an integer linear program which is dealt with via relaxation and rounding.

Ruggeri and Saupe [RS08] propose an algorithm which estimates the difference between distance matrices derived from surfaces by comparing their histograms of inter point distances. The theoretical foundations for this method can be found in [BK07]. There, it is shown that most weighted graphs can be reconstructed from their distribution of weights and the distribution of sums of weights of adjacent edges.

In this section, we describe an algorithmic framework which performs very well in practice. At first, we investigate a simple *point signature* for general metric spaces — so-called *local distance sets* — in Section 3.3.1. Furthermore, we present exact algorithms (Section 3.3.2) for the problems related to the Gromov-Hausdorff distance mentioned in Section 3.2. Next, we develop the theoretical foundations and algorithms to drastically reduce the complexity of the problem, by identifying a small number of points in the metric spaces which capture their essential structure. A correspondence between those points can be extended to a correspondence between the whole spaces with low error, see Section 3.3.3. These methods can also be used for partial non-rigid shape matching, which is discussed in the last part of this chapter.

As mentioned above, a straightforward way to compare metric spaces is by examining the distances occurring in the spaces. Ruggeri and Saupe [RS08] compare the histograms of inter point distances of both spaces. To achieve a a meaningful comparison, the metric spaces must have the same cardinality and the points should be "evenly distributed": consider the metric

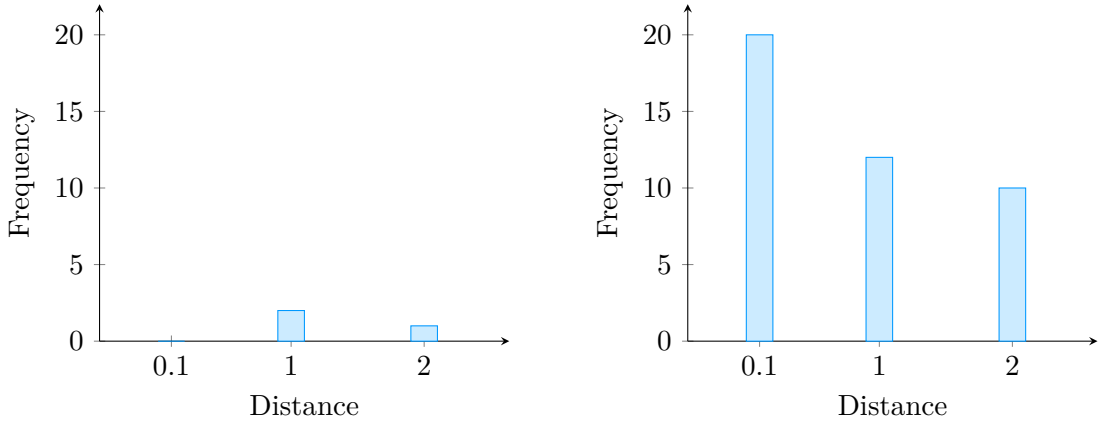
spaces

$$X := \{-1, 0, 1\} \subseteq \mathbb{R}$$

for $N \in \mathbb{N}$ and

$$Y := \{-1, 0, 1 - \varepsilon/1, \ldots, 1 - \varepsilon/N, 1, 1 + \varepsilon/N, \ldots, 1 + \varepsilon/1\} \subseteq \mathbb{R}$$

with metrics induced by the Euclidean distances. We have $d_{GH}(X, Y) = \varepsilon$, but there is no obvious measure subject to which the histograms of distances reflect this similarity, cf. Figure 3.8.



(a) Histogram of distances of (X, d_X).

(b) Histogram of distances of (Y, d_Y), for $N = 2$ and $\varepsilon = 0.05$.

Figure 3.8: Histograms of distances of metric spaces with low Gromov-Hausdorff distance are not necessarily similar.

Another approach with more discriminatory power is to consider distance histograms of each point and to match vertices with similar such histograms. This is similar to attempts in the literature to estimate the similarity of shapes by comparing point signatures.

3.3.1 Local Distance Sets

To identify corresponding points in surfaces, several methods for computing local characteristics of a point on a surface —so-called point signatures — for 2-dimensional Riemannian manifolds have been considered. Sun, Ovsjanikov, and Guibas [SOG09] use properties of the heat diffusion process on a Riemannian manifold to derive a multi-scale point signature — referred to as heat kernel signature — which is invariant with respect to isometries. Other signatures with this property have also been considered [GSCO07; Rus07].

Since we are concerned with the Gromov-Hausdorff distance of arbitrary metric spaces, we do not consider these signatures for Riemannian manifolds in the following and concentrate on properties of general metric spaces. Local distance sets have been considered by Grigorescu and Petkov [GP03] and Mémoli [Mé12].

Definition 3.30 (Local distance set)
Let (X, d_X) be a metric space and $n \coloneqq |X|$. The local distance set at $x \in X$ is the ordered multiset

$$D_X(x) \coloneqq (d_x^1, \ldots, d_x^{n-1}) \subseteq \mathbb{R}_{\geq 0}$$

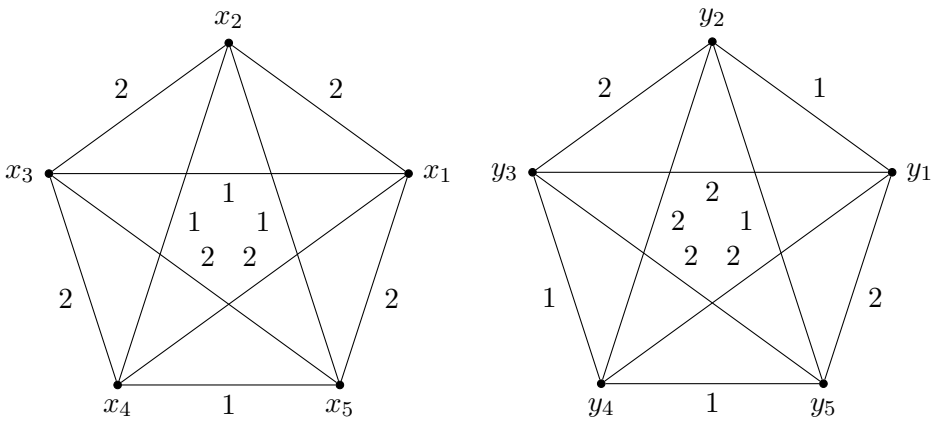
where $d_x^i \coloneqq d_X(x, x_i)$ for some $x_i \in X \setminus \{x\}, i \in \{1, \ldots, n\}$ and $d_x^i \leq d_x^j$ for all $i \leq j$.

Unfortunately, this approach is not sufficient to distinguish metric spaces in general, see Figure 3.9, and also failed in several practical examples investigated.

However, [Mé12] shows that the Hausdorff distance of local distance sets can be used to obtain a lower bound for the Gromov-Hausdorff distance of two metric spaces: for metric spaces (X, d_X), (Y, d_Y), we have

$$d_{GH}(X, Y) \geq \frac{1}{2} \inf_{R \in \mathcal{R}(X,Y)} \sup_{(x,y) \in R} d_H(D_X(x), D_Y(y)).$$

We generalize this observation to the (bottleneck) p-Gromov-Hausdorff distance in Corollary 3.34, but first establish a related instrumental property of local distance sets.



(a) (X, d_X) **(b)** (Y, d_Y)

Figure 3.9: Local distance sets are not sufficient to characterize metric spaces: (X, d_X) and (Y, d_Y) have identical sets of local distance sets; the local distance set of x_1, y_1, x_3 and y_3 is $(1, 2, 2, 2)$ while x_2, y_2, x_4, y_4, x_5 and y_5 have local distance set $(1, 1, 2, 2)$ in their respective metric spaces. However, the Gromov-Hausdorff distance of (X, d_X) and (Y, d_Y) is $1/2$.

Local distance sets can be used as point signatures which enable the a priori exclusion of assignments which necessarily induce correspondences with high distortion. To compare these distance sets, we introduce the histogram distance:

Definition 3.31 (Histogram distance)
Let (X, d_X) and (Y, d_Y) be finite metric spaces and $p \in \mathbb{N} \cup \{\infty\}$. The histogram distance of $x \in X$ and $y \in Y$ is given by

$$d_{HI,p}(x, y) \coloneqq \frac{1}{2} \min_{R_H \in \mathcal{R}(D_X(x), D_Y(y))} \sqrt[p]{\frac{1}{(|X| + |Y|)^2} \sum_{(d_x, d_y) \in R_H} |d_x - d_y|^p}$$

for $p \in \mathbb{N}$ and

$$d_{HI,\infty}(x,y) \coloneqq \frac{1}{2} \min_{R_H \in \mathcal{R}(D_X(x),D_Y(y))} \max_{(d_x,d_y)\in R_H} |d_x - d_y|.$$

The bottleneck histogram distance can analogously be defined as

$$d_{BHI,p}(x,y) \coloneqq \frac{1}{2} \min_{R_H \in \Pi(D_X(x),D_Y(y))} \sqrt[p]{\frac{1}{(|X|+|Y|)^2} \sum_{(d_x,d_y)\in R_H} |d_x - d_y|^p}$$

for $p \in \mathbb{N}$ and

$$d_{BHI,\infty}(x,y) \coloneqq \frac{1}{2} \min_{R_H \in \Pi(D_X(x),D_Y(y))} \max_{(d_x,d_y)\in R_H} |d_x - d_y|.$$

It is easy to see that $d_{HI,p} \leq d_{BHI,p}$ for all $p \in \mathbb{N} \cup \{\infty\}$.

For $p = \infty$, the (bottleneck) histogram distance is the Hausdorff distance of the local distance sets:

Remark 3.32
We have

$$d_{HI,\infty}(x,y) = d_H(D_X(x), D_X(y))$$

and similarly

$$d_{BHI,\infty}(x,y) = d_B(D_X(x), D_X(y)),$$

i. e., the bottleneck histogram distance is given by the bottleneck distance of the local distance sets, cf. Lemma 1.6. Both can be easily computed, see, e. g., [EI96].

As mentioned, the histogram distance can be used to identify assignments which cannot be contained in a relation R with small distortion. This is crucial for the exact algorithms presented in the following, since it allows to reduce the complexity of the input considerably.

Lemma 3.33
Let (X, d_X) and (Y, d_Y) be finite metric spaces, $p \in \mathbb{N} \cup \{\infty\}$, and $R \in \mathcal{R}(X,Y)$ a correspondence. For $x \in X$, $y \in Y$, and $(x,y) \in R$ we have

$$d_{HI,p}(x,y) \leq \frac{1}{2}\Gamma_p(R).$$

Similarly, for a bottleneck correspondence $R \in \Pi(X,Y)$ with $(x,y) \in R$ we have

$$d_{BHI,p}(x,y) \leq \frac{1}{2}\Gamma_p(R).$$

Proof. We show $d_{HI,p}(x,y) \leq \frac{1}{2}\Gamma_p(R)$. The proof for the other cases is analogous.

Given $R \in \mathcal{R}(X,Y)$, a correspondence $R_H \in \mathcal{R}(D_X(x), D_Y(y))$ between $D_X(x)$ and $D_Y(y)$ is induced by R in the following way: For each $x' \in X$ there exists $y' \in Y$ such that $(x',y') \in R$ and similarly for each $y' \in Y$ there is $x' \in X$ with $(x',y') \in R$. Hence

$$R_H \coloneqq \{(d_X(x,x'), d_Y(y,y')) \in \mathbb{R} \times \mathbb{R} \mid (x',y') \in R\}$$

is a correspondence in $\mathcal{R}(D_X(x), D_Y(y))$. With this correspondence, we have

$$d_{HI,p}(x,y) \leq \frac{1}{2}\sqrt[p]{\frac{1}{(|X|+|Y|)^2} \sum_{(d_x,d_y)\in R_H} |d_x - d_y|^p} \leq \frac{1}{2}\Gamma_p(R) \qquad \square$$

A lower bound for the p-Gromov-Hausdorff distance of two metric spaces in terms of the histogram distance is an immediate consequence:

Corollary 3.34
Let (X, d_X) and (Y, d_Y) be finite metric spaces and $p \in \mathbb{N} \cup \{\infty\}$. We have

$$\min_{R \in \mathcal{R}(X,Y)} \max_{(x,y) \in R} d_{HI,p}(x, y) \leq d_{GH,p}(X, Y)$$

and

$$\min_{R \in \Pi(X,Y)} \max_{(x,y) \in R} d_{BHI,p}(x, y) \leq d_{BGH,p}(X, Y).$$

In contrast to the (bottleneck) Gromov-Hausdorff distance, the bottleneck histogram distance can be efficiently computed. Before giving an optimal bottleneck correspondence in Theorem 3.36, we prove a technical supporting lemma.

Lemma 3.35
Let $x_1, x_2, y_1, y_2 \in \mathbb{R}$ be real numbers with $x_2 \geq x_1$ and $y_2 \geq y_1$. For any $p \in \mathbb{N}$ we have

$$|x_2 - y_1|^p + |x_1 - y_2|^p \geq |x_2 - y_2|^p + |x_1 - y_1|^p$$

and

$$\max\{|x_2 - y_1|, |x_1 - y_2|\} \geq \max\{|x_2 - y_2|, |x_1 - y_1|\}.$$

Proof. For the first inequality, let $a \coloneqq x_1 - y_1$, $b \coloneqq x_2 - y_2$ and $\delta_x \coloneqq x_2 - x_1$. Considering the symmetry of the inequality, we can assume $a \geq 0$ (otherwise exchange x and y). The constraint $x_2 \geq x_1$ can be rewritten as $\delta_x \geq 0$ and $y_2 \geq y_1$ can be reformulated as $0 \leq y_2 - y_1 = a - b + \delta_x$, which is equivalent to $\delta_x \geq b - a$. The inequality in question can then be obtained from the inequality

$$|a + \delta_x|^p + |b - \delta_x|^p \geq |a|^p + |b|^p \text{ for } a, b, \delta_x \in \mathbb{R}, a \geq 0, \delta_x \geq \max\{0, b - a\}$$

by substitution, and is proved in the following. For given $a, b \in \mathbb{R}$, $a \geq 0$, consider the real function

$$f : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R},\ f(\delta_x) \coloneqq |a + \delta_x|^p + |b - \delta_x|^p.$$

It is continuous on $\mathbb{R}_{\geq 0}$ and continuously differentiable for $\delta_x \in \mathbb{R}_{>0} \setminus \{b\}$ (since $a + \delta_x \geq a \geq 0$, i. e., $-a \notin \mathbb{R}_{\geq 0}$) with derivative

$$f'(\delta_x) = p\Big((a + \delta_x)^{p-1} + \operatorname{sign}(b - \delta_x)^p (b - \delta_x)^{p-1}\Big).$$

To prove the correctness of the inequality, we show

$$\min_{\substack{\delta_x \geq b - a \\ \delta_x \geq 0}} f(\delta_x) \geq |a|^p + |b|^p.$$

First, we consider the case $b \geq a \geq 0$. In this case, we have

$$a + \delta_x \geq a + (b - a) = b \geq b - \delta_x$$

and

$$a + \delta_x \geq \delta_x \geq \delta_x - b = -(b - \delta_x),$$

i. e., $a + \delta_x \geq |b - \delta_x|$ and therefore $f'(\delta_x) \geq 0$. Therefore, the minimal value of f for $\delta_x \geq b - a$ is attained at $\delta_x \coloneqq b - a$ and $f(b - a) = |a|^p + |b|^p$.

In case $b < a$, consider first $-b > a \geq 0$. We have $|a + \delta_x| = a + \delta_x \geq |a|$ and additionally $|b - \delta_x| = \delta_x - b \geq -b = |b|$. In consequence, the inequality holds.

Otherwise, i. e., if $-b \leq a$, we have

$$a + \delta_x \geq a > b \geq b - \delta_x$$

and

$$a + \delta_x \geq -b + \delta_x = \delta_x - b,$$

i. e., $a + \delta_x \geq |b - \delta_x|$. This implies $f'(\delta_x) \geq 0$ for $\delta_x \geq 0$, and the minimal value of f for $\delta_x \geq 0$ is therefore attained at $\delta_x \coloneqq 0$ with $f(0) = |a|^p + |b|^p$, which proves the claim.

For the second inequality, we first show

$$|x_1 - y_1| \leq \max\{|x_1 - y_2|, |x_2 - y_1|\}.$$

Since we assume (without loss of generality) $x_1 \geq y_1$, we have

$$|x_1 - y_1| = x_1 - y_1 \leq x_2 - y_1 = |x_2 - y_1|.$$

To show that

$$|x_2 - y_2| \leq \max\{|x_1 - y_2|, |x_2 - y_1|\}$$

holds, we consider the case $x_2 < y_2$, which implies

$$|x_2 - y_2| = y_2 - x_2 \leq y_2 - x_1 = |x_1 - y_2|$$

and $x_2 \geq y_2$, yielding

$$|x_2 - y_2| = x_2 - y_2 \leq x_2 - y_1 = |x_2 - y_1|. \qquad \square$$

The next theorem states that the p-norm of the vector of differences induced by a correspondence between two sets of real numbers is optimal if it respects the ordering of the sets. This property can easily be used to construct an optimal correspondence between ordered sets such as local distance sets.

Theorem 3.36
Let $(X, d_X), (Y, d_Y)$ be metric spaces with $n \coloneqq |X| = |Y|$ and $x \in X$, $y \in Y$. Consider the local distance sets $D_X(x) \coloneqq (d_x^1, \ldots, d_x^{n-1})$ and $D_Y(y) \coloneqq (d_y^1, \ldots, d_y^{n-1})$ with $d_x^1 \leq \ldots \leq d_x^{n-1}$ and $d_y^1 \leq \ldots \leq d_y^{n-1}$. Then

$$R_H \coloneqq \left\{(d_x^i, d_y^i) \mid i \in \{1, \ldots, n-1\}\right\} \in \Pi(D_X(x), D_Y(y))$$

is an optimal bottleneck correspondence with respect to the bottleneck histogram distance, for any $p \in \mathbb{N} \cup \{\infty\}$.

Proof. Consider an arbitrary bottleneck correspondence $R \in \Pi(D_X(x), D_Y(y))$. A *crossing* of R is given by indices $1 \leq i < j \leq n-1$ and $1 \leq k < l \leq n-1$ such that $(d_x^i, d_y^l), (d_x^j, d_y^k) \in R$, compare Figure 3.10. Let R_C be an optimal bottleneck correspondence with a minimal number of crossings and assume $R_C \neq R_H$. Then R_C must contain a crossing $1 \leq i < j \leq n-1$, $1 \leq k < l \leq n-1$, and Lemma 3.35 states that

$$\left|d_x^i - d_y^k\right|^p + \left|d_x^j - d_y^l\right|^p \leq \left|d_x^i - d_y^l\right|^p + \left|d_x^j - d_y^k\right|^p$$

for $p \in \mathbb{N}$ and

$$\max\left\{\left|d_x^i - d_y^k\right|, \left|d_x^j - d_y^l\right|\right\} \leq \max\left\{\left|d_x^i - d_y^l\right|, \left|d_x^j - d_y^k\right|\right\}$$

for $p = \infty$, i. e., replacing the assignments $(d_x^i, d_y^l), (d_x^j, d_y^k)$ in R_C with $(d_x^i, d_y^k), (d_x^j, d_y^l)$ yields a correspondence $R'_C \in \Pi(X, Y)$ with $\Gamma(R'_C) \leq \Gamma(R_C)$ and fewer crossings, which is a contradiction. □

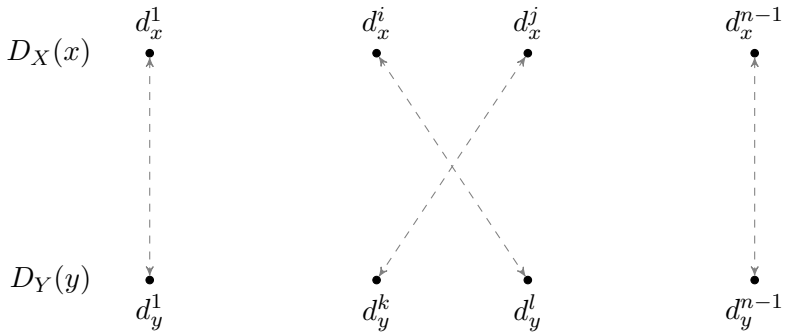


Figure 3.10: There is an optimal correspondence between local distance sets which does not contain crossings.

As mentioned above, see Lemma 3.33, these point signatures are an efficient instrument to determine tuples of points which cannot be matched in a correspondence with low distortion. This is of great help for the exact algorithm presented in the next section, as it allows for a preprocessing step that significantly decreases the input sizes of this algorithm in practice.

3.3.2 Exact Algorithms

In the following, we give integer linear programming formulations for the Problems GHD, p-GHD and p-BGHD introduced in Section 3.2 on finite metric spaces (X, d_X), (Y, d_Y) with $X \coloneqq \{x_1, \ldots, x_m\}$, $Y \coloneqq \{y_1, \ldots, y_n\}$. We also show that the optimization version of Problem δ-p-SGHD for $p = \infty$ can be recast in the form of the maximum clique problem (Problem MCQ). For this problem, sophisticated algorithms which work well on reasonably sized instances are known [Bom+99; Ös02].

To see the similarity between these problems, recall that in order to solve Problem δ-p-SGHD, we have to find a relation $R \subseteq X \times Y$ such that $\Gamma_{ijkl} \coloneqq \Gamma(\{(x_i, y_k), (x_j, y_l)\}) \leq \delta$ for all $(x_i, y_k), (x_j, y_l) \in R$ and $|R| \geq N$, or prove that no such R exists. This can be solved by the

integer linear program (ILP)

$$\begin{aligned} \max \quad & \sum_{i=1}^{m}\sum_{j=1}^{n}\xi_{ik} \\ \text{s.t.} \quad & \xi_{ik}+\xi_{jl}\leq 1 && \forall i,j\in\{1,\dots,m\}, k,l\in\{1,\dots,n\} \text{ with } \Gamma_{ijkl}>\delta \\ & \xi_{ik}\in\{0,1\} && i\in\{1,\dots,m\}, k\in\{1,\dots,n\}. \end{aligned}$$

The binary variables ξ_{ik}, $i\in\{1,\dots,m\}, k\in\{1,\dots,n\}$ represent whether (x_i, y_k) is in R or not, i. e., for a solution $\xi\in\{0,1\}^{m\times n}$ the corresponding relation is given by

$$R_\xi \coloneqq \{(x_i,y_k) \mid \xi_{ik}=1\} \text{ with } |R_\xi| = \sum_{i=1}^{m}\sum_{k=1}^{n}\xi_{ik}.$$

This is exactly the ILP formulation for Problem MCQ. Consider the graph $G\coloneqq(V,E)$, where

$$V \coloneqq \left\{(x_i,y_k)\in X\times Y \mid d_{HI,\infty}(x_i,y_k)\leq\frac{1}{2}\delta\right\}$$

is the set of all possible pairs of points of X and Y which can be contained in a correspondence with distortion no more than δ, see Lemma 3.33. The set of edges is given by

$$E\coloneqq\{\{(x_i,y_k),(x_j,y_l)\} \mid (x_i,y_k),(x_j,y_j)\in V,\ \Gamma_{ijkl}\leq\delta\}.$$

The construction is similar to the so-called modular product of graphs, which is used to transform the maximum common subgraph isomorphism problem to Problem MCQ, see [BB76]. Two assignments $(x_i,y_k),(x_j,y_l)\in V$ are connected by the edge $\{(x_i,y_k),(x_j,y_l)\}$ if and only if $\Gamma_{ijkl}=\Gamma(\{(x_i,y_k),(x_j,y_l)\})\leq\delta$. This means that two assignments can be in the same relation only if the distances of the points in the respective metric spaces are within the specified tolerance δ. An implementation in pseudo-code is given in Algorithm 3.1.

Input: Rational metric spaces $(X,d_X),(Y,d_Y)$, upper bound on their Gromov-Hausdorff distance $\delta\geq 0$
Output: Relation $R\subseteq X\times Y$ of maximal cardinality with $\Gamma(R)\leq\delta$

1 $V\leftarrow\left\{(x_i,y_k)\in X\times Y \mid d_{HI,\infty}(x_i,y_k)\leq\frac{1}{2}\delta\right\}$
2 $E\leftarrow\{\{(x_i,y_k),(x_j,y_l)\} \mid (x_i,y_k),(x_j,y_j)\in V,\ \Gamma_{ijkl}\leq\delta\}$
3 $G\leftarrow(V,E)$
4 $R\leftarrow$ `maximumClique(G)`

Algorithm 3.1: computeSGHD

Furthermore, it is straightforward to adapt this approach to Problem δ-p-SBGHD. In this modified model, the vertex set is given by

$$V \coloneqq \left\{(x_i,y_k)\in X\times Y \mid d_{BHI,\infty}(x_i,y_k)\leq\frac{1}{2}\delta\right\}$$

and two assignments (x_i,y_k), $(x_j,y_l)\in V$ are connected only if they do not share a common point, i. e., $x_i\neq x_j$ and $y_k\neq y_l$, in addition to incurring a distortion less than δ. The bottleneck

Gromov-Hausdorff distance is less than or equal to $\delta/2$ if and only if there exists a clique of size $|X| = |Y|$ in the corresponding graph.

The maximum clique problem is a fundamental problem in combinatorial optimization and has therefore been subject to a substantial amount of research. While the problem is — as mentioned in the previous section — hard to approximate, there are algorithms which yield exact solutions for practical instances in a reasonable running time, see [Ös02]. In our experiments, we could solve instances with up to 60 points in each metric space.

The Gromov-Hausdorff distance of (X, d_X) and (Y, d_Y) can also be computed exactly by the integer linear program

$$\begin{aligned}
&\min && \delta && \\
&\text{s.t.} && \sum_{k=1}^{n} \xi_{ik} \geq 1 && \forall i \in \{1, \dots, m\} \\
& && \sum_{i=1}^{m} \xi_{ik} \geq 1 && \forall k \in \{1, \dots, n\} \\
& && \xi_{ik} + \xi_{jl} \leq 1 + \delta\gamma_{ijkl} && \forall i, j \in \{1, \dots, m\}, \forall k, l \in \{1, \dots, n\} \\
& && \xi_{ik} \in \{0, 1\} && \forall i \in \{1, \dots, m\}, \quad \forall k \in \{1, \dots, n\}.
\end{aligned}$$

Again, $\xi_{ik} \in \{0, 1\}$ indicates whether $x_i \in X$ is matched to $y_k \in Y$ in an optimal correspondence $R \in \mathcal{R}(X, Y)$, i. e.,

$$R_\xi \coloneqq \{(x_i, y_k) \in X \times Y \mid \xi_{ik} = 1\}$$

and

$$\gamma_{ijkl} \coloneqq \frac{1}{\Gamma_{ijkl}} = \frac{1}{\Gamma((x_i, y_j), (x_k, y_l))}$$

denotes the inverse of the distortion induced by two assignments.

Similarly, Problem p-GHD can also be computed using an integer linear program:

$$\begin{aligned}
z = &\min \sum_{i,j=1}^{m} \sum_{k,l=1}^{n} (\Gamma_{ijkl})^p \zeta_{ijkl} && \\
&\text{s.t.} && \sum_{i=1}^{m} \xi_{ik} \geq 1 && \forall k \in \{1, \dots, n\} \\
& && \sum_{k=1}^{n} \xi_{ik} \geq 1 && \forall i \in \{1, \dots, m\} \\
& && \zeta_{ijkl} \geq \xi_{ik} + \xi_{jl} - 1 && \\
& && \zeta_{ijkl} \in \{0, 1\} && \forall i, j \in \{1, \dots, m\}, \forall k, l \in \{1, \dots, n\} \\
& && \xi_{ik} \in \{0, 1\} && \forall i \in \{1, \dots, m\}, \quad \forall k \in \{1, \dots, n\},
\end{aligned}$$

In addition to the variables used in the previous program, we introduce the decision variables $\zeta_{ijkl} \in \{0, 1\}$. These indicate whether both x_i is matched to y_k and x_j is matched to y_l, such that the distortion induced by these assignments can be summed up in the objective function. The p-Gromov Hausdorff-distance is then given by

$$d_{GH,p}(X, Y) = \frac{1}{2} \sqrt[p]{\frac{z}{(n+m)^2}}.$$

Both ILP formulations can be easily adapted to compute an optimal bottleneck correspondence by requiring equality in the "assignment constraints", i. e.,

$$\sum_{i=1}^{m} \xi_{ik} = 1 \qquad \forall k \in \{1, \dots, n\}$$
$$\sum_{k=1}^{n} \xi_{ik} = 1 \qquad \forall i \in \{1, \dots, m\}.$$

Unfortunately, the LP-relaxation of first formulation is rather weak and the number of binary variables in the second formulation is huge. Therefore a solution via standard ILP solvers is generally only possible for very small instances.

Problem p-BGHD can also be reformulated as another well-known problem in graph theory. Similar to the graph constructed for the clique problem, we consider the complete graph $G := (V, E)$ on the vertex set $V := X \times Y$ with edge weights Γ_{ijkl} for $i \neq k, j \neq l$, and $\Gamma_{ijkl} := M$ for $M \in \mathbb{N}$ with $M \geq \operatorname{diam}(X) + \operatorname{diam}(Y)$ otherwise. The problem is now equivalent to finding a subset of $n := |X| = |Y|$ nodes of G which induce a subgraph of minimal weight.

Since we search for an induced subgraph on exactly n nodes, i. e., with exactly $\frac{n(n-1)}{2}$ edges, this can be transformed into a dense k-subgraph problem by using the weight function $\hat{\Gamma} := M - \Gamma$. A more thorough treatment of this problem is given in Chapter 4.

Since our focus is mainly on the Gromov-Hausdorff distance of metric spaces which are rather large in practice, we also investigate methods to reduce the size of an instance.

3.3.3 Extension of a Relation

Usually, shapes in practice have few distinctive feature points and it is intuitively clear that a relation between these feature points suffices to construct a correspondence with low distortion of all points. The example shown in Figure 3.11 depicts two shapes with highlighted feature points. The methods described in this section suffice to extend a relation between these feature points to a correspondence, while keeping the distortion low.

Similar to the approach we used in Chapter 2, we try to identify *bases*, i. e., small subsets of metric spaces, such that a correspondence between bases of two spaces naturally induces a correspondence between the spaces themselves.

It is well-known that each point in Euclidean 3-space is uniquely characterized by its distances to 4 arbitrary affinely independent fixed points, see e. g., [Dat05]. This fact, often referred to as trilateration or triangulation [HB01], is used in manifold positioning applications in practice, such as satellite navigation [Tho98], location services for mobile devices based on wireless connections [Pat+03; Nag04] and many others.

For the problem at hand, we introduce a generalization of this concept to metric spaces. We are interested in subsets of a metric space for which points can be, at least approximately, distinguished by their distances to the points in these subsets.

Definition 3.37 (Base of a metric space)
Let (X, d_X) be a metric space and $\varepsilon \geq 0$. A subset $B \subseteq X$ is called ε-base if for each $x, x' \in X$ there exists $b \in B$ such that

$$d_X(x, x') \leq \varepsilon + |d_X(x, b) - d_X(x', b)|.$$

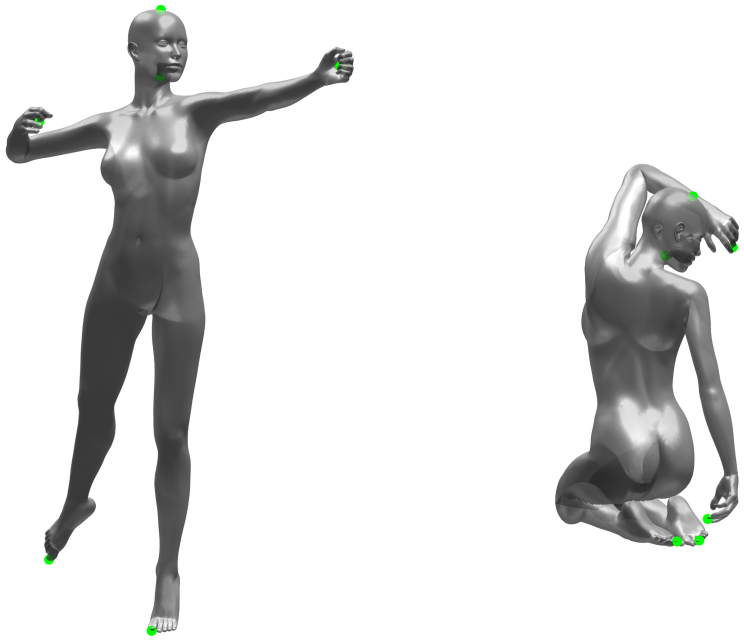


Figure 3.11: Feature points of a shape in different poses, which allow for a correspondence which can be extended with low error.

For finite X, Y and $B \subseteq X$, we denote by $\varepsilon_X(B) \geq 0$ the minimal real value such that B is an $\varepsilon_X(B)$-base.

Intuitively, Definition 3.37 requires that two points with similar distances to all points in the base cannot be far apart. Therefore, the distances to points in the base can be considered as "coordinates" of a point. As an example, consider Figure 3.11. By specifying the distances of a point on the surface to all highlighted feature points, its position can be located with small error. Another example is given in Figure 3.12, in which $\{x_4, x_5\}$ is a 0-base.

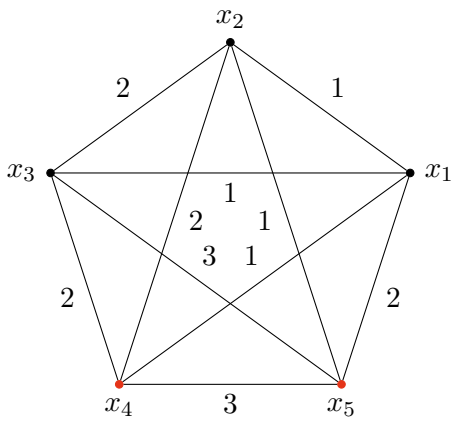


Figure 3.12: Example of a metric space with small 0-base $\{x_4, x_5\}$ depicted in red.

Next, we observe that the distance between two points is determined (up to an error of ε)

by these coordinates.

Lemma 3.38
For an ε-base B of a metric space (X, d_X), we have

$$\sup_{b\in B}|d_X(x,b) - d_X(x',b)| \leq d_X(x,x') \leq \varepsilon + \sup_{b\in B}|d_X(x,b) - d_X(x',b)|$$

for all $x, x' \in X$ due to the triangle inequality.

This is also similar in spirit to sparse sampling and matrix completion [CT10], where one tries to reconstruct a matrix of low rank from a fraction of the entries: for a finite metric space (X, d_X) with an ε-base $B \subseteq X$, the distance matrix D_X can be approximately reconstructed from the knowledge of the columns (or rows) of D_X which correspond to the points of B.

Note, however, that the distance matrices we consider here can in general not be approximated by low rank matrices within an acceptable tolerance, even if they admit small ε-bases: Consider the metric space depicted in Figure 3.12. The corresponding distance matrix

$$D_X = \begin{pmatrix} 0 & 1 & 1 & 1 & 2 \\ 1 & 0 & 2 & 2 & 1 \\ 1 & 2 & 0 & 2 & 3 \\ 1 & 2 & 2 & 0 & 3 \\ 2 & 1 & 3 & 3 & 0 \end{pmatrix}$$

has full rank, but the metric space contains a 0-base of cardinality 2, as shown above. Furthermore, the metric spaces depicted in Figures 3.9, 3.13 and 3.19 all have full rank, but have smaller 0-bases.

Unfortunately, unlike the bases used in Chapter 2, where the existence of wide bases with size bounded by the dimension of the surrounding space is proven in Theorem 2.7, the size of the ε-bases defined here cannot be easily bounded for fixed ε:

Consider the discrete metric space (X, d_D), where $d_D(x, x') := 1$ for all $x, x' \in X$ with $x \neq x'$, and $d_D(x, x') := 0$ otherwise. It is easy to see that this space does not contain an ε-base of size less than $|X| - 1$ for $\varepsilon < 1$. Note that the metric spaces constructed in the proof of Theorem 3.24 are constructed in a similar way, and also do not admit an ε-base of size less than $|X| - 1$ for sufficiently small ε, but can be considered as discretizations of manifolds embedded in $\mathbb{R}^2$.

An important feature of ε-bases is that this property is, at least approximately, conserved by correspondences.

Lemma 3.39
Let (X, d_X), (Y, d_Y) be two metric spaces and $B_X \subseteq X$ an ε_X-base of X. Let $R \in \mathcal{R}(X, Y)$ be a correspondence, $R_B \subseteq R$ a relation with $\mathcal{P}_1(R_B) = B_X$, and let $B_Y := \mathcal{P}_2(R_B)$. Then B_Y is an ε_Y-base of Y with $\varepsilon_Y \leq \varepsilon_X + 3\Gamma(R)$, and this bound is tight.

Proof. Let $y, y' \in Y$. Since R is a correspondence, there exist $x, x' \in X$ with $(x, y), (x', y') \in R$. As B_X is an ε_X-base of X, there exists $b_x \in B_X$ such that

$$d_X(x, x') \leq \varepsilon_X + |d_X(x, b_x) - d_X(x', b_x)|.$$

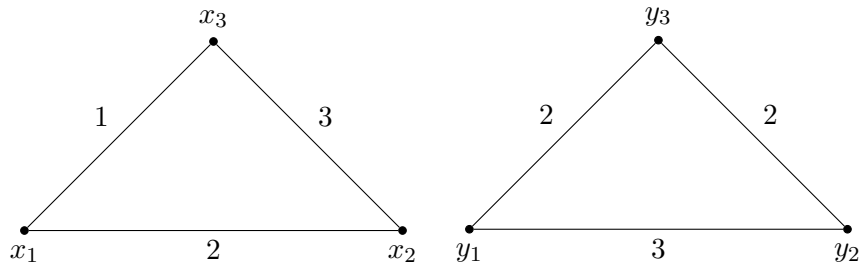
Furthermore, there exists $b_y \in B_Y$ such that $(b_x, b_y) \in R_B$. We use

$$\begin{aligned}\left|d_X(x,b_x) - d_X(x',b_x)\right| &\leq \left|d_X(x,b_x) - d_Y(y,b_y)\right| + \left|d_Y(y,b_y) - d_Y(y',b_y)\right| \\ &\quad + \left|d_Y(y',b_y) - d_X(x',b_x)\right| \\ &\leq 2\Gamma(R) + \left|d_Y(y,b_y) - d_Y(y',b_y)\right|,\end{aligned}$$

to show

$$\begin{aligned}d_Y(y,y') &\leq d_X(x,x') + \Gamma(R) \\ &\leq \varepsilon_X + \Gamma(R) + \left|d_X(x,b_x) - d_X(x',b_x)\right| \\ &\leq \varepsilon_X + 3\Gamma(R) + \left|d_Y(y,b_y) - d_Y(y',b_y)\right|,\end{aligned}$$

which concludes the proof.



(a) (X, d_X) **(b)** (Y, d_Y)

Figure 3.13: Worst case example for the bound on bases given in Lemma 3.39.

For the tightness of the bound, consider the metric spaces depicted in Figure 3.13. An optimal correspondence is given by $R \coloneqq \{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ with $\Gamma(R) = 1$ and $B_X \coloneqq \{x_3\}$ is a 0-base of X. The corresponding set $B_Y \coloneqq \{y_3\}$ is not an ε-base of Y for $\varepsilon < 3$. □

The generalization of the localization of a point by its distances to the points of an ε-base introduced in Definition 3.37 and Lemma 3.38 can now be used to extend a correspondence between subspaces to cover the whole spaces.

Definition 3.40 (Extension of a relation)
Let (X, d_X) and (Y, d_Y) be compact metric spaces and $R_B \subseteq X \times Y$ a relation. A correspondence $R \in \mathcal{R}(X, Y)$ such that $R_B \subseteq R$ is called extension of R_B to $\mathcal{R}(X, Y)$.

A greedy extension of R_B is defined by

$$\begin{aligned}R_G \coloneqq R_B \cup &\left\{(x,y) \in X \times Y \mid x \in X \setminus \mathcal{P}_1(R_B),\ y = \operatorname*{argmin}_{1,\, y' \in Y} \sup_{(b_x,b_y)\in R_B} \left|d_X(x,b_x) - d_Y(y',b_y)\right|\right\} \\ \cup &\left\{(x,y) \in X \times Y \mid y \in Y \setminus \mathcal{P}_2(R_B),\ x = \operatorname*{argmin}_{1,\, x' \in X} \sup_{(b_x,b_y)\in R_B} \left|d_X(x',b_x) - d_Y(y,b_y)\right|\right\}\end{aligned}$$

and is denoted by $\mathcal{E}(R_B)$. The error occurring in the extension is given by

$$\begin{aligned}\zeta(R_B) \coloneqq \max \Bigg\{ &\max_{x\in X} \min_{y\in Y} \sup_{(b_x,b_y)\in R_B} |d_X(x,b_x) - d_Y(y,b_y)|, \\ &\max_{y\in Y} \min_{x\in X} \sup_{(b_x,b_y)\in R_B} |d_X(x,b_x) - d_Y(y,b_y)| \Bigg\}.\end{aligned}$$

It is easy to see that a greedy extension is in fact an extension of R_B.

Lemma 3.41
In the setting of Definition 3.40, $R_G \coloneqq \mathcal{E}(R_B)$ is an extension of R_B.

Proof. The greedy extension is well-defined: an elementary argument shows that the function

$$f : X \times Y \longrightarrow \mathbb{R}_{\geq 0},\ f(x,y) \coloneqq \sup_{(b_x,b_y)\in R_B} |d_X(x,b_x) - d_Y(y,b_y)|$$

is continuous on $\{x\} \times Y$ for fixed $x \in X$ and on $X \times \{y\}$ for fixed $y \in Y$. Since $\{x\} \times Y$ and $X \times \{y\}$ are compact, the existence of $\operatorname{argmin}_{1y'\in Y} f(x,y')$ and $\operatorname{argmin}_{1x'\in X} f(x',y)$ is guaranteed. Furthermore, we have $R_B \subseteq R_G$ and $X = \mathcal{P}_1(R_G)$, $Y = \mathcal{P}_2(R_G)$ by definition. □

Combining the above definitions, we are able to bound the error introduced by a greedy extension of a correspondence.

Lemma 3.42
Let (X, d_X), (Y, d_Y) be two compact metric spaces with ε-bases B_X and B_Y, $R_B \in \mathcal{R}(B_X, B_Y)$ a correspondence, and $R_G \coloneqq \mathcal{E}(R_B) \in \mathcal{R}(X,Y)$ a greedy extension. We have

$$\Gamma(R_G) \leq \varepsilon + 2\zeta(R_B).$$

Proof. Let $(x,y),(x',y') \in R_G$. We can assume $d_X(x,x') \geq d_Y(y,y')$ without loss of generality. Using Lemma 3.38, we have

$$\begin{aligned}
d_X(x,x') - d_Y(y,y') &\leq \varepsilon + \sup_{b_x\in B_X} \big|d_X(x,b_x) - d_X(x',b_x)\big| - d_Y(y,y') \\
&\leq \varepsilon + \sup_{b_x\in B_X} \big|d_X(x,b_x) - d_X(x',b_x)\big| \\
&\quad - \sup_{b_y\in B_Y} \big|d_Y(y,b_y) - d_Y(y',b_y)\big| \\
&\leq \varepsilon + \sup_{(b_x,b_y)\in R_B} \Big|\big|d_X(x,b_x) - d_X(x',b_x)\big| \\
&\qquad\qquad - \big|d_Y(y,b_y) - d_Y(y',b_y)\big|\Big| \\
&\leq \varepsilon + \sup_{(b_x,b_y)\in R_B} \Big|d_X(x,b_x) - d_X(x',b_x) \\
&\qquad\qquad - d_Y(y,b_y) + d_Y(y',b_y)\Big| \\
&\leq \varepsilon + \sup_{(b_x,b_y)\in R_B} \Big(|d_X(x,b_x) - d_Y(y,b_y)| \\
&\qquad\qquad + \big|d_X(x',b_x) - d_Y(y',b_y)\big|\Big) \\
&\leq \varepsilon + 2\zeta(R_B).
\end{aligned}$$

□

Lemma 3.42 shows that the distortion of a greedy extension of a correspondence between ε-bases can be bounded by the maximal error occurring in the extension. In fact, one can prove that a greedy extension is a 2-approximation of an optimal extension for $\varepsilon = 0$.

Theorem 3.43
Let (X, d_X), (Y, d_Y) be two compact metric spaces with ε-bases B_X and B_Y, respectively. Let $R_B \in \mathcal{R}(B_X, B_Y)$ be a correspondence and $R_G \coloneqq \mathcal{E}(R_B)$ a greedy extension according to Definition 3.40. Then

$$\Gamma(R_G) \leq \varepsilon + 2\zeta(R_B) \leq \varepsilon + 2 \inf_{\substack{R \in \mathcal{R}(X,Y) \\ R_B \subseteq R}} \Gamma(R)$$

and this bound is tight.

Proof. Let $R \in \mathcal{R}(X, Y)$ be a correspondence with $R_B \subseteq R$. Using Lemma 3.42, we have $\Gamma(R_G) \leq \varepsilon + 2\zeta(R_B)$. Further, we have $\zeta(R_B) \leq \Gamma(R)$, as we will see in the following. Let $(x, y) \in R_G$ and assume

$$y = \operatorname*{argmin}_{\tilde{y} \in Y} \sup_{(b_x, b_y) \in R_B} |d_X(x, b_x) - d_Y(\tilde{y}, b_y)|.$$

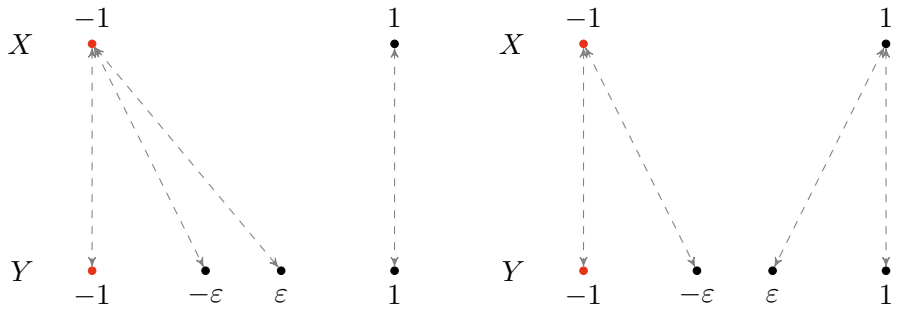
Since R is a correspondence, there is $\hat{y} \in Y$ such that $(x, \hat{y}) \in R$. For arbitrary $\varepsilon' > 0$, there exists $(b_x, b_y) \in R_B \subseteq R$ such that

$$|d_X(x, b_x) - d_Y(y, b_y)| \leq |d_X(x, b_x) - d_Y(\hat{y}, b_y)| + \varepsilon'$$

Analogous arguments can be applied for the cases $(x, y) \in R_B$ and $(x, y) \in R_G$ with

$$x = \operatorname*{argmin}_{\tilde{x} \in X} \sup_{(b_x, b_y) \in R_B} |d_X(\tilde{x}, b_x) - d_Y(y, b_y)|.$$

This implies $\zeta(R_B) \leq \Gamma(R)$.



(a) Optimal extension R_{GH} of R_B. **(b)** Greedy extension R_G of R_B.

Figure 3.14: Worst case error of a greedy extension of a correspondence.

To prove the tightness of the bound, consider the metric spaces depicted in Figure 3.14 as subspaces of $\mathbb{R}$ for arbitrary $0 < \varepsilon < 1$. The sets $B_X \coloneqq \{-1\}$ and $B_Y \coloneqq \{-1\}$ (depicted in red) are 0-bases of X and Y, respectively, and $R_B \coloneqq \{(-1, -1)\} \in \mathcal{R}(B_X, B_Y)$ is a correspondence between them. The correspondence R_{GH}, depicted as dashed lines in Figure 3.14a, is an optimal extension of R_B with distortion $\Gamma(R_{GH}) = 1 + \varepsilon$. The greedy extension $R_G \coloneqq \mathcal{E}(R_B)$ of R_B has distortion $\Gamma(R_G) = 2(1 - \varepsilon)$, cf. Figure 3.14b. □

However, the Gromov-Hausdorff distance of two metric spaces X and Y cannot be bounded in terms of the Gromov-Hausdorff distance of bases $B_X \subseteq X$ and $B_Y \subseteq Y$. Assume X and Y are discretizations of the same manifold with differing different sampling densities, e. g., consider the metric spaces depicted in Figure 3.15. The points of the sets depicted in red are 0-bases with Gromov-Hausdorff distance 0, but the Gromov-Hausdorff distance of the sets is $\alpha/2$, and therefore arbitrary.

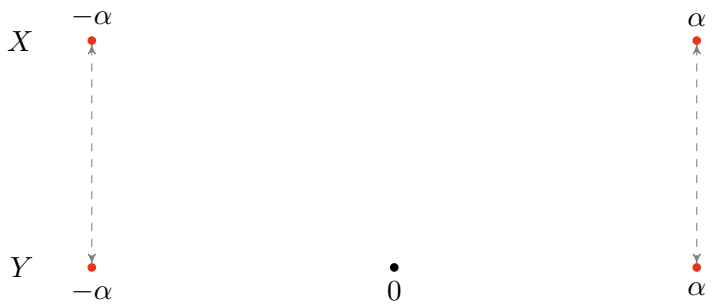


Figure 3.15: The Gromov-Hausdorff distance cannot be bounded by the Gromov-Hausdorff distance of bases (depicted in red).

Still, we can prove a result similar to Theorem 2.13.

Theorem 3.44
Let (X, d_X), (Y, d_Y) be compact metric spaces and B_X an ε_X-base of X. For arbitrary $\varepsilon > 0$, there exists a base $B_Y \subseteq Y$ of Y and a correspondence $R_B \in \mathcal{R}(B_X, B_Y)$ such that

$$d_{GH}(X, Y) \leq \frac{1}{2}\Gamma(\mathcal{E}(R_B)) \leq \frac{\varepsilon_X}{2} + 5d_{GH}(X, Y) + \varepsilon.$$

Proof. For a correspondence $R_o \in \mathcal{R}(X, Y)$ with $\Gamma(R_o) \leq 2d_{GH}(X, Y) + \frac{\varepsilon}{5}$, let

$$R_B \coloneqq \{(x, y) \in R_o \mid x \in B_X\} \text{ and } B_Y \coloneqq \mathcal{P}_2(R_B).$$

Lemma 3.39 guarantees that B_Y is an ε_Y-base of Y with

$$\varepsilon_Y \leq \varepsilon_X + 3\Gamma(R_o) = \varepsilon_X + 6d_{GH}(X, Y) + \frac{3}{5}\varepsilon.$$

Theorem 3.43 implies that the distortion of a greedy extension $R_G \coloneqq \mathcal{E}(R_B)$ of R_B is bounded by

$$\begin{aligned}\Gamma(R_G) &\leq \max\{\varepsilon_X, \varepsilon_Y\} + 2\Gamma(R_o) \\ &\leq \varepsilon_X + 10d_{GH}(X, Y) + \varepsilon.\end{aligned}$$

□

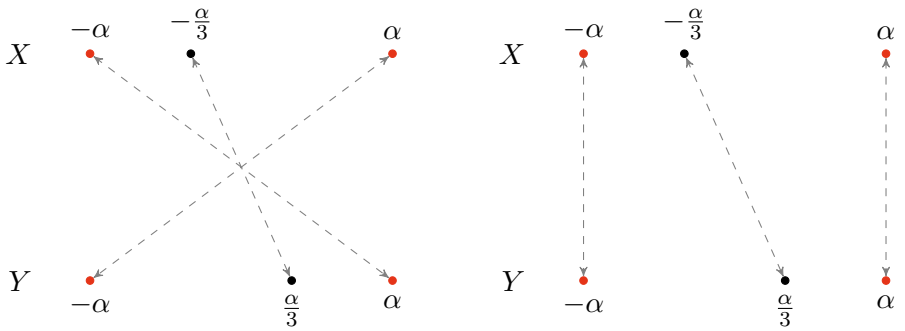
These results can be combined to obtain an algorithm similar in spirit to Algorithm 2.1 in Chapter 2. At first, we find a base B_X of X and then evaluate optimal correspondences between B_X and B_Y for every $B_Y \subseteq Y$ with $|B_Y| = |B_X|$. Note that we are not interested in matching a subset of X to a subset of Y, i. e., we do not need to consider all possible subsets B_X of X.

Input: Rational metric spaces (X, d_X), (Y, d_Y), $\delta \geq 0$
Output: Correspondence $R \subseteq X \times Y$, distortion γ

```
1  B_X ← findBase(P)
2  γ ← ∞
3  for all B_Y ⊆ Q with |B_Y| = |B_X| do
4  |  R' ← computeSGHD((X, d_X), (Y, d_Y), B_X, B_Y, δ)
5  |  R' ← extendCorrespondence((X, d_X), (Y, d_Y), R)
6  |  if Γ(R') < γ then
7  |  |  R ← R'
8  |  |  γ ← Γ(R')
9  |  end
10 end
```

Algorithm 3.2: nonrigidBasisMatching

In contrast to the situation in Chapter 2, we cannot guarantee that this algorithm yields an approximation of the optimal solution. Consider the metric spaces depicted in Figure 3.16, where a possible solution computed by Algorithm 3.2 is depicted. While the metric spaces X and Y are isometric, the distortion of the computed correspondence is $2/3\,\alpha$ for an arbitrary $\alpha \geq 0$, although the bases are chosen as in the proof of Theorem 3.44. The problem occurs since Algorithm 3.2 computes an arbitrary optimal correspondence (we do not consider ordered bases here!) between the bases, while the proof of the theorem requires a specific correspondence between the bases. This could be remedied by considering all relations between B_X and B_Y instead of an optimal correspondence, but the computational cost is prohibitive in practice.



(a) Correspondence R_{GH} with distortion $\Gamma(R_{GH}) = 0$.

(b) Greedy (and optimal) extension R_G with distortion $\Gamma(R_G) = 2/3\,\alpha$.

Figure 3.16: A greedy or even optimal extension of an optimal correspondence between bases (depicted in red) can yield a correspondence with arbitrarily high distortion.

Even Algorithm 3.2 is not suitable for practical instances of greater size, since an $\mathcal{NP}$-hard problem has to be solved in each iteration.

In the following, we try to further reduce the number of bases to consider and develop an algorithm which is able to determine whether a relation can be extended with little error or not.

3.3.4 Extreme Points

An approach to reduce the number of bases to consider is to require certain additional properties. This idea can also be transferred from Euclidean geometry, where *extreme points* suffice to characterize the class of compact convex sets. In case of metric spaces, a generalization of extreme points can be defined.

Definition 3.45 (Extreme point)
Let (X, d_X) be a metric space and $\gamma, \eta \geq 0$. A point $x \in X$ is called (γ, η)-extreme point of X if

$$d_X(x_1, x_2) < d_X(x_1, x) + d_X(x, x_2) - \gamma$$

for all $x_1, x_2 \in X \setminus \mathbb{B}_x(\eta) = \{x' \in X \mid d_X(x', x) > \eta\}$.

The definition requires that going through an extreme point x is a detour of length at least γ for any source and destination point which are not in the vicinity of x. All feature points highlighted in Figure 3.11 are extreme points for suitable γ and η, since they are at "remote" locations of the shape.

In case that (X, d_X) is an induced subspace of an Euclidean space, the extreme points are just $(0, 0)$-extreme points of the associated metric space.

Unfortunately, it is not always possible to find extreme points that also form an ε-base for small ε. Consider, for example, the unit sphere $X \coloneqq \{x \in \mathbb{R}^2 \mid \|x\| = 1\}$ with the metric induced by geodesic distances. This metric space does not have any (γ, η)-extreme points, for any $\gamma \geq 0, \eta \in [0, \frac{\pi}{2})$. For $\eta \geq \frac{\pi}{2}$, every point $x \in X$ is $(0, \eta)$-extreme. In many practical examples however, a set of extreme points for suitable parameters is also an ε-base with small ε.

Similar to the property of bases shown in Lemma 3.39, extreme points are also approximately maintained by a correspondence.

Lemma 3.46
Let (X, d_X), (Y, d_Y) be two metric spaces, $R \in \mathcal{R}(X, Y)$ a correspondence, and $x \in X$ a (γ_X, η_X)-extreme point of X. Each $y \in Y$ with $(x, y) \in R$ is a (γ_Y, η_Y)-extreme point of Y if $0 \leq \gamma_Y \leq \gamma_X - 3\Gamma(R)$ and $\eta_Y \geq \eta_X + \Gamma(R)$.

Proof. Let $y \in Y$ with $(x, y) \in R$. If $Y \setminus \mathbb{B}_y(\eta_Y) = \emptyset$, then y is a (γ, η_Y) extreme point for any $\gamma \geq 0$. Otherwise, consider $y_1, y_2 \in Y \setminus \mathbb{B}_y(\eta_Y)$. Since R is a correspondence, there exist $x_1, x_2 \in X$ such that $(x_1, y_1), (x_2, y_2) \in R$. We have

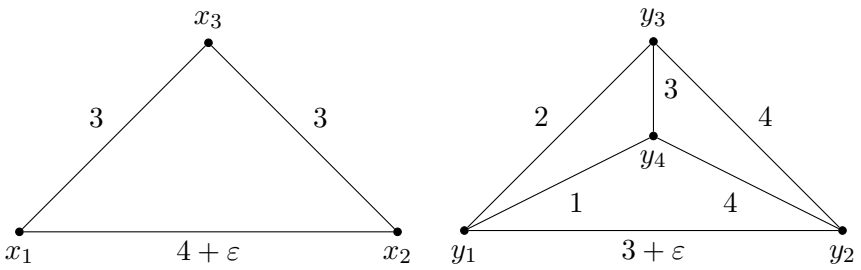
$$d_X(x_i, x) \geq d_Y(y_i, y) - \Gamma(R) > \eta_Y - \Gamma(R) \geq \eta_X$$

for $i \in \{1, 2\}$, i. e., $x_1, x_2 \in X \setminus \mathbb{B}_x(\eta_X)$. Using the fact that x is (γ_X, η_X)-extreme, we have

$$\begin{aligned} d_Y(y_1, y_2) &\leq d_X(x_1, x_2) + \Gamma(R) \\ &< d_X(x_1, x) + d_X(x, x_2) - \gamma_X + \Gamma(R) \\ &\leq d_Y(y_1, y) + d_Y(y, y_2) - (\gamma_X - 3\Gamma(R)), \end{aligned}$$

which implies that $y \in Y$ is (γ_Y, η_Y)-extreme in Y.

As an example of metric spaces X and Y where this bound is best possible, consider the metric spaces given in Figure 3.17 for $0 < \varepsilon < 1$ and the (optimal) correspondence $R \coloneqq \{(x_1, y_1), (x_1, y_4), (x_2, y_2), (x_3, y_3)\}$ with $\Gamma(R) = 1$. The point x_1 is $(4, 0)$-extreme in X while the corresponding point y_1 is $(1, 1)$-extreme in Y but not (γ, η)-extreme for any $(\gamma, \eta) \in ([1 + \varepsilon, \infty) \times [0, 2)) \cup ([0, 1) \times [0, 1))$. □



(a) (X, d_X) **(b)** (Y, d_Y)

Figure 3.17: Worst case example for the bound on extreme points given in Lemma 3.46.

We can now combine our results to give an approximation algorithm for Problem GHD.

Theorem 3.47
Let (X, d_X), (Y, d_Y) be two finite metric spaces and $m \coloneqq |X|$, $n \coloneqq |Y|$. Let $B_X \subseteq X$ be a set of (γ_X, η_X)-extreme points in X and $B_Y \subseteq Y$ be the set of all (γ_Y, η_Y)-extreme points in Y, where $\gamma_X \geq 0, \eta_X \geq 0$, $0 \leq \gamma_Y \leq \gamma_X - 6d_{GH}(X,Y)$, and $\eta_Y \geq \eta_X + 2d_{GH}(X,Y)$. We can compute a correspondence $R \in \mathcal{R}(X,Y)$ with

$$d_{GH}(X,Y) \leq \frac{1}{2}\Gamma(R) \leq \frac{\varepsilon_X(B_X)}{2} + 5d_{GH}(X,Y)$$

in time $\mathcal{O}\Big(2^{kl}(m^2+n^2)(k+l)\Big)$ for $k \coloneqq |B_X|$, $l \coloneqq |B_Y|$, i. e., the problem is fixed parameter tractable with respect to the parameter kl.

Proof. We evaluate all relations $R_B \subseteq B_X \times B_Y$. Each such correspondence R_B is assigned a score

$$\mathcal{S}(R_B) \coloneqq \max\{\varepsilon_X(\mathcal{P}_1(R_B)), \varepsilon_Y(\mathcal{P}_2(R_B))\} + 2\zeta(R_B).$$

Let $R_{\mathcal{S}} \in \operatorname{argmin}_{R_B \subseteq B_X \times B_Y} \mathcal{S}(R_B)$. By Theorem 3.43, we know that the distortion of a greedy extension $R_G \coloneqq \mathcal{E}(R_{\mathcal{S}})$ of $R_{\mathcal{S}}$ is bounded by

$$\Gamma(R_G) \leq \max\{\varepsilon_X(\mathcal{P}_1(R_{\mathcal{S}})), \varepsilon_Y(\mathcal{P}_2(R_{\mathcal{S}}))\} + 2\zeta(R_{\mathcal{S}}) = \mathcal{S}(R_{\mathcal{S}}).$$

Let $R_o \in \mathcal{R}(X,Y)$ be an arbitrary correspondence with minimal distortion. Consider a (γ_X, η_X)-extreme point $x \in B_X$ and a point $y \in Y$ such that $(x,y) \in R_o$. Lemma 3.46 and the definition of γ_Y and η_Y ensure that y is (γ_Y, η_Y)-extreme, i. e., $y \in B_Y$. Therefore, the relation

$$R_B \coloneqq \{(x,y) \in R_o \mid x \in B_X\}$$

satisfies $\mathcal{P}_1(R_B) = B_X$ and $\mathcal{P}_2(R_B) \subseteq B_Y$. Lemma 3.39 shows $\varepsilon_Y(B_Y) \leq \varepsilon_X(B_X) + 3\Gamma(R_o)$ and Theorem 3.43 yields $\mathcal{S}(R_B) \leq \max\{\varepsilon_X(\mathcal{P}_1(R)), \varepsilon_Y(\mathcal{P}_2(R))\} + 2\Gamma(R_o)$.

Combining the mentioned partial results and using $\Gamma(R_o) = 2d_{GH}(X,Y)$ yields

$$\Gamma(R_G) \leq \mathcal{S}(R_{\mathcal{S}}) \leq \mathcal{S}(R_B) \leq \varepsilon_X(B_X) + 10d_{GH}(X,Y).$$

The running time of the algorithm is determined by the evaluation of all relations $R_B \subseteq B_X \times B_Y$, which are at most 2^{kl}. It is sufficient to consider relations R_B with $|R_B| \leq k+l$, see Lemma 1.7. The computational cost of computing of $\zeta(R)$ is in $\mathcal{O}(nm|R_B|)$ and $\varepsilon_X(\mathcal{P}_1(R_B))$ and $\varepsilon_Y(\mathcal{P}_2(R_B))$ can be determined in $\mathcal{O}(m^2k)$ and $\mathcal{O}(n^2l)$, respectively. □

Since the complexity of computing a correspondence between bases is determined almost solely by their size, we need to be able to precisely control the number of points in the bases. Suitably adjusting the parameters γ and η of extreme points is not easy — and sometimes not feasible, see the example after Definition 3.45 — in practice. Furthermore, it is easy to see that all points in a sufficiently small neighbourhood (depending on γ and η) of an extreme point are extreme points as well.

Lemma 3.48
Let (X, d_X) be a metric space, $x \in X$ a (γ, η)-extreme point for $\gamma, \eta \geq 0$, and $x' \in \mathbb{B}_x(\varepsilon)$ for $\varepsilon \geq 0$ such that $\gamma - 2\varepsilon \geq 0$. Then x' is a $(\gamma - 2\varepsilon, \eta + \varepsilon)$-extreme point.

Proof. Due to the triangle inequality, we have $\mathbb{B}_x(\eta) \subseteq \mathbb{B}_{x'}(\eta + \varepsilon)$. Consider the points $x_1, x_2 \in X \setminus \mathbb{B}_{x'}(\eta + \varepsilon) \subseteq X \setminus \mathbb{B}_x(\eta)$. Since x is (γ, η)-extreme, we have

$$d_X(x_1, x') + d_X(x', x_2) \geq d_X(x_1, x) - 2\varepsilon + d_X(x, x_2) > d_X(x_1, x_2) + \gamma - 2\varepsilon,$$

i. e., x' is $(\gamma - 2\varepsilon, \eta + \varepsilon)$-extreme. □

As an example, consider the metric space given in Figure 3.18. The points x_3 and x_4 are both $(2, \varepsilon)$-extreme. Since the points are close to each other, it is not necessary to consider both points to approximately describe the essential structure of the metric space.

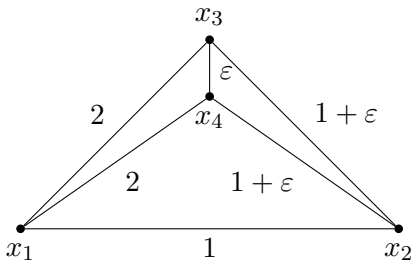


Figure 3.18: Points in the neighbourhood of an extreme point are also extreme points, cf. Lemma 3.48.

In practice, this problem is also prevalent. Consider, e. g., the feature point on the woman's toe in Figure 3.11. It is obvious that this point and all points in a reasonably small neighbourhood (depending on the parameters γ and η) are extreme points. However, for an approximation of the shape matching problem, we are satisfied with matching the toe to a suitable part in a corresponding shape. Thus, for a relation which can be extended with low error, it is sufficient to include only one of these points.

Therefore, we are also interested in the slightly modified concept of *extreme sets*, which allow us to specify the size of the bases we want to consider and do not contain multiple points of each "cluster" of extreme points.

Definition 3.49 (Extreme set)
Let (X, d_X) be a metric space, $\gamma > 0$ and $E \subseteq X$. E is called γ-extreme set if for each $x \in X$ there exist $x_1, x_2 \in E$ such that

$$d_X(x_1, x) + d_X(x, x_2) \leq \gamma + d_X(x_1, x_2).$$

Informally, the definition of an extreme set requires that for each point $x \in X$ there are two points $x_1, x_2 \in E$ in the extreme set, such that going from x_1 to x_2 over x is not a big detour.

This ensures that no point in X is "very extreme" with respect to E and therefore all "clusters" of extreme points are represented by E for a given accuracy.

For the metric space given in Figure 3.18, $\{x_1, x_3\}$ and $\{x_1, x_3, x_4\}$ are ε-extreme sets, while $\{x_3, x_4\}$ is not γ-extreme for any $\gamma \leq 4 - \varepsilon$.

As with bases, it is not possible to bound the error induced by extending an optimal correspondence between extreme sets. Figure 3.19 shows a metric space (X, d_X) which contains the 0-extreme sets $B_1 \coloneqq \{x_2, x_3, x_5\}$ and $B_2 \coloneqq \{x_1, x_3, x_5\}$ which are also 0-bases. An optimal correspondence between these sets is given by $R_B \coloneqq \{(x_2, x_5), (x_3, x_1), (x_5, x_3)\}$ with distortion $\Gamma(R_B) = 0$. Furthermore, the local distance sets of all $x \in X \setminus \{x_4\}$ are equal, i. e., R_B additionally respects the local distance sets. $R_G \coloneqq \{(x_2, x_5), (x_3, x_1), (x_5, x_3), (x_4, x_4), (x_1, x_2)\}$ is a greedy extension of R_B with $\Gamma(R_G) = 1$. Theorem 3.43 implies that there cannot be an extension of R_B with distortion 0, while obviously $d_{GH}(X, X) = 0$.

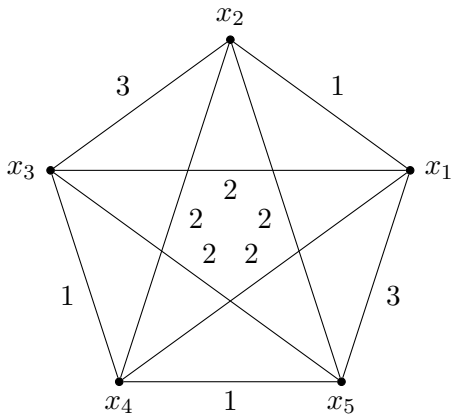


Figure 3.19: The error induced by extending an optimal correspondence between extreme sets cannot be bounded: $\{x_2, x_3, x_5\}$ and $\{x_1, x_3, x_5\}$ are 0-extreme sets and 0-bases of the depicted metric space and $\{(x_2, x_5), (x_3, x_1), (x_5, x_3)\}$ is a correspondence with distortion 0 between them which also respects local distance sets. However, it cannot be extended to a correspondence of the whole space without distortion.

In general, extreme sets must not necessarily induce ε-bases with small ε. An example is given in Figure 3.20.

However, extreme sets work extremely well for selecting relevant feature points on practical examples (see Figures 3.11, 3.21 and 3.22). Furthermore, a relation between extreme sets can often be extended to a near optimal correspondence. This is demonstrated in the following.

3.3.5 Implementation

To achieve good results on practical instances in an acceptable running time, efficient techniques for the reduction of the size of the instances are essential. At first, rather small extreme sets of both shapes are computed greedily, see Algorithm 3.3. In each iteration, we determine the point $x \in X \setminus E$ which is "most extreme" with respect to E (line 3) and add it to the extreme set E.

Lemma 3.50
Let (X, d_X) be a finite metric space with cardinality $m \coloneqq |X|$ and $s_e \in \mathbb{N}$. Algorithm 3.3 computes an extreme set of size s_e in time $\mathcal{O}(s_e^3 m + m^2)$.

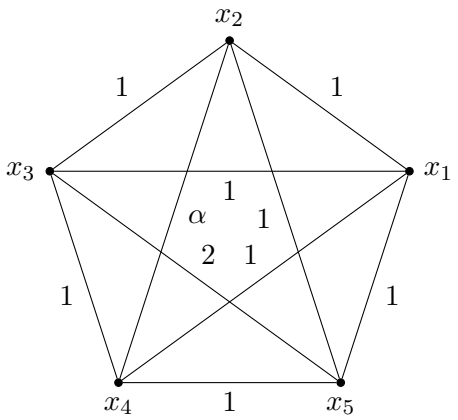


Figure 3.20: Extreme sets do not necessarily induce ε-bases with small ε: $\{x_3, x_5\}$ is 0-extreme, but no ε-base for $\varepsilon < \alpha$, $1 < \alpha < 2$; x_3, x_5 are $(2 - \alpha - \eta, 0)$ extreme for $0 < \eta < 2 - \alpha$.

We proceed by computing an optimal solution to Problem δ-p-SGHD, using the metric spaces induced by the extreme sets and an estimate δ for the minimal distortion as input for Algorithm 3.1.

Input: Metric space (X, d_X), size of extreme set $s_e \in \mathbb{N}$
Output: $\gamma \geq 0$ and γ-extreme set E

$E \leftarrow \{x_1, x_2\}$ for diameter inducing points $x_1, x_2 \in X$
for $k \in \{3, \ldots, s_e\}$ **do**
  $e \leftarrow \operatorname{argmax}_1 \min_{x \in X \setminus E \quad e_1, e_2 \in E} d_X(e_1, x) + d_X(x, e_2) - d_X(e_1, e_2)$
  $E \leftarrow E \cup \{e\}$
end
$\gamma \leftarrow \max_{x \in X \setminus E} \min_{e_1, e_2 \in E} d_X(e_1, x) + d_X(x, e_2) - d_X(e_1, e_2)$

Algorithm 3.3: extremeSet

If the estimate for the minimal distortion δ is good enough, we obtain a correspondence with low distortion between two subsets of the extreme sets which are ε-bases with small ε. This correspondence is then extended greedily, see Definition 3.40. An exemplary version of this algorithm in pseudo-code is shown in Algorithm 3.4.

Lemma 3.51
Let (X, d_X), (Y, d_Y) be finite metric spaces with $m := |X|$, $n := |Y|$ and $R_B \subseteq X \times Y$ a relation. Algorithm 3.4 computes a greedy extension R_G of R_B in time $\mathcal{O}(nm|R_B|)$.

As we have seen in the preceding part, the extension of a correspondence can introduce an error of at most a factor of 2 in addition to the additive term ε. Reconsidering the worst-case example from Figure 3.14 and the proof of Theorem 3.43, it is evident that this is caused by focusing solely on the distances of a point to the base and neglecting points which have already been included in the correspondence. Algorithm 3.5 tries to mitigate this problem by taking all currently added points into account for the search of a suitable match $y \in Y$ and

Input: Metric spaces $(X, d_X), (Y, d_Y)$, bases $B_X \subseteq X$, $B_Y \subseteq Y$, correspondence $R_B \in \mathcal{R}(B_X, B_Y)$
Output: Greedy extension $R_G \coloneqq \mathcal{E}(R_B) \in \mathcal{R}(X, Y)$, $\zeta \geq 0$

```
1  ζ ← 0
2  R_G ← R_B
3  for x ∈ X \ P_1(R_G) do
4  |  y* ← argmin_1 max_{y∈Y} max_{(x',y')∈R_B} |d_X(x, x') − d_Y(y, y')|
5  |  c ← max_{(x',y')∈R_B} |d_X(x, x') − d_Y(y*, y')|
6  |  R_G ← R_G ∪ {(x, y*)}
7  |  ζ ← max{c, ζ}
8  end
9  for y ∈ Y \ P_2(R_G) do
10 |  x* ← argmin_1 max_{x∈X} max_{(x',y')∈R_B} |d_X(x, x') − d_Y(y, y')|
11 |  c ← max_{(x',y')∈R_B} |d_X(x*, x') − d_Y(y, y')|
12 |  R_G ← R_G ∪ {(x*, y)}
13 |  ζ ← max{c, ζ}
14 end
```

Algorithm 3.4: extendCorrespondence

$x \in X$ for all points $x' \in X' \subseteq X$ and $y' \in Y' \subseteq Y$, respectively. Thus, the correspondence is incrementally extended to cover at least X' and Y'. Setting $X' \coloneqq X$ and $Y' \coloneqq Y$ results in a correspondence $R \in \mathcal{R}(X, Y)$. It is easy to see that this does not enhance the approximation ratio in theory, but yields improved results for most instances in practice.

Unfortunately, this algorithm is computationally much more expensive for larger sets X' and Y', since all assignments $(x, y) \in R$ have to be included in the evaluation of an additional assignment in the later stages of the algorithm.

Lemma 3.52
Let (X, d_X), (Y, d_Y) be finite metric spaces with $m \coloneqq |X|$, $n \coloneqq |Y|$, $R_B \subseteq X \times Y$ a relation and $X' \subseteq X, Y' \subseteq Y$. For $k \coloneqq |X'|$, $l \coloneqq |Y'|$, Algorithm 3.5 computes a relation R with $R_B \subseteq R$, $X' \subseteq \mathcal{P}_1(R)$ and $Y' \subseteq \mathcal{P}_2(R)$ in time $\mathcal{O}((kn + lm)(k + l + |R_B|))$.

Proof. We have $|R| \leq k + l + |R_B|$ by construction, which yields the result. □

For the problem sizes we consider in practice, this can lead to running times which are in general not acceptable. Therefore, we also implemented a multi-resolution approach. In Algorithm 3.6, we first compute sufficiently dense subspaces $X' \subseteq X$ and $Y' \subseteq Y$ of a prescribed size s_r of both spaces with $\mathcal{P}_1(R_B) \subseteq X'$ and $\mathcal{P}_2(R_B) \subseteq Y'$. Algorithm 3.5 is used to compute a relation covering all points of X' and Y'. This relation is then extended using Algorithm 3.4.

As a last step, we can now iteratively refine the resulting correspondence R_G. The improved correspondence is initialized by $R_I \leftarrow R_G$. Algorithm 3.7 determines an assignment $(\hat{x}, \hat{y}) \in R_I$ inducing the distortion of the improved correspondence R_I (line 3) and evaluates several options for improvement. At first, the assignment $(\hat{x}, \hat{y})$ is removed from R_I. If $\hat{x}$ is no longer covered

Input: Metric spaces $(X, d_X), (Y, d_Y)$, $B_X \subseteq X$, $B_Y \subseteq Y$ and correspondence $R_B \in \mathcal{R}(B_X, B_Y)$, $X' \subseteq X, Y' \subseteq Y$
Output: Relation R with $R_B \subseteq R$, $X' \subseteq \mathcal{P}_1(R)$, $Y' \subseteq \mathcal{P}_2(R)$, $\zeta \geq 0$

$\zeta \leftarrow 0$
$R \leftarrow R_B$
for $x \in X' \setminus \mathcal{P}_1(R)$ **do**
  $y^* \leftarrow \operatorname{argmin}_1 \max_{(x',y') \in R} |d_X(x, x') - d_Y(y, y')|$
  $c \leftarrow \max_{(x',y') \in R} |d_X(x, x') - d_Y(y^*, y')|$
  $R \leftarrow R \cup \{(x, y^*)\}$
  $\zeta \leftarrow \max\{c, \zeta\}$
end
for $y \in Y' \setminus \mathcal{P}_2(R)$ **do**
  $x^* \leftarrow \operatorname{argmin}_1 \max_{(x',y') \in R} |d_X(x, x') - d_Y(y, y')|$
  $c \leftarrow \max_{(x',y') \in R} |d_X(x^*, x') - d_Y(y, y')|$
  $R \leftarrow R \cup \{(x^*, y)\}$
  $\zeta \leftarrow \max\{c, \zeta\}$
end

Algorithm 3.5: incExtendCorrespondence

Input: Metric spaces $(X, d_X), (Y, d_Y)$, $B_X \subseteq X$, $B_Y \subseteq Y$, correspondence $R_B \in \mathcal{R}(B_X, B_Y)$ and size of reference set s_r
Output: Greedy extension $R_G = \mathcal{E}(R_B) \in \mathcal{R}(X, Y)$

Compute evenly distributed subspace X' of X with $|X'| = s_r$ and $\mathcal{P}_1(R_B) \subseteq X'$
Compute evenly distributed subspace Y' of Y with $|Y'| = s_r$ and $\mathcal{P}_2(R_B) \subseteq Y'$
$R \leftarrow$ `incExtendCorrespondence`$((X, d_X), (Y, d_Y), R_B, X', Y')$
$R_G \leftarrow$ `extendCorrespondence`$((X, d_X), (Y, d_Y), R)$
$R_I \leftarrow$ `improveCorrespondence`$((X, d_X), (Y, d_Y), R_G, 10^{-5})$

Algorithm 3.6: extendCorrespondenceRef

by R_I, i. e., there exists no $y' \in Y$ with $(\hat{x}, y') \in R_I$, we compensate by computing $y^* \in Y$ such that the distortion of $R_I \cup \{(\hat{x}, y^*)\}$ is minimized and set $R_I \leftarrow R_I \cup \{(\hat{x}, y^*)\}$. We proceed analogously in case y is no longer covered. In case both $\hat{x}$ and $\hat{y}$ are already matched, the assignment $(\hat{x}, \hat{y})$ can be removed without substitute. If the resulting correspondence has higher distortion than the preceding correspondence, we retract the changes. This process is iterated until the improvement of a single substitution falls below a specified tolerance τ. In conjunction with the aforementioned algorithm, this produces results comparable to Algorithm 3.5 in practice, while being significantly faster.

Input: Metric spaces $(X, d_X), (Y, d_Y)$ and correspondence $R_G \in \mathcal{R}(X, Y)$, tolerance $\tau > 0$
Output: Improved correspondence $R_I \in \mathcal{R}(X, Y)$

$R_I \leftarrow R_G$
repeat
  $R \leftarrow R_I$
  $(\hat{x}, \hat{y}) \leftarrow \operatorname{argmax}_{(x,y)\in R_I} \max_{(x',y')\in R_I} |d_X(x, x') - d_Y(y, y')|$
  $R_I \leftarrow R_I \setminus \{(\hat{x}, \hat{y})\}$
  if $\hat{x} \notin \mathcal{P}_1(R_I)$ **then**
    $y^* \leftarrow \operatorname{argmin}_{y'\in Y} \max_{(x,y)\in R_I} |d_X(\hat{x}, x) - d_Y(y', y)|$
    $R_I \leftarrow R_I \cup \{(\hat{x}, y^*)\}$
  end
  if $\hat{y} \notin \mathcal{P}_2(R_I)$ **then**
    $x^* = \operatorname{argmin}_{x'\in X} \max_{(x,y)\in R_I} |d_X(x', x) - d_Y(\hat{y}, y)|$
    $R_I \leftarrow R_I \cup \{(x^*, \hat{y})\}$
  end
  $\iota \leftarrow \Gamma(R) - \Gamma(R_I)$
  if $\iota < 0$ **then**
    $R_I \leftarrow R$
  end
until $\iota > \tau$

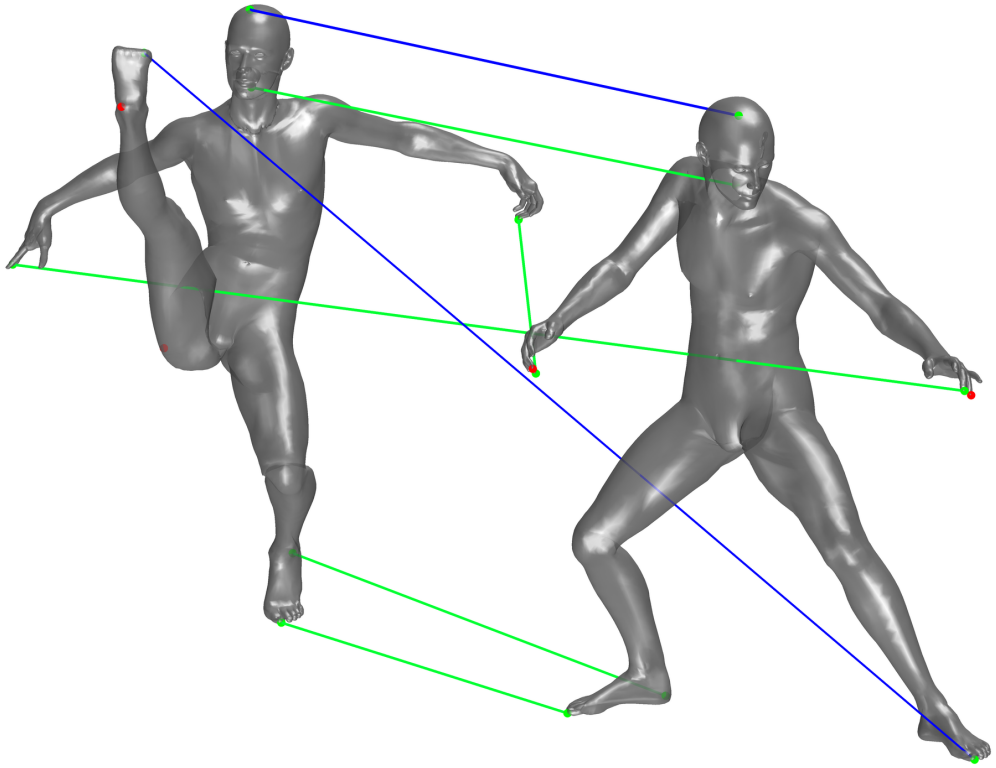
Algorithm 3.7: improveCorrespondence

3.3.6 Results

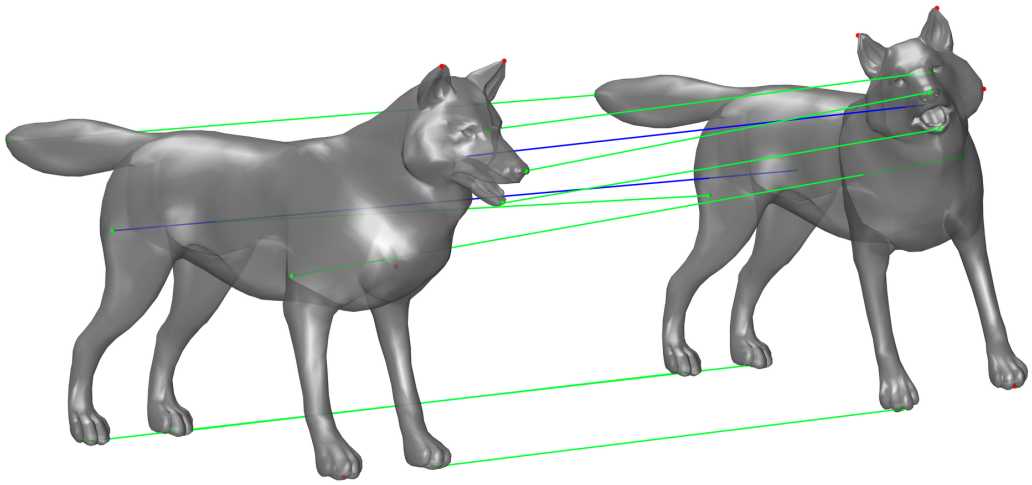
The algorithms discussed in the previous section can now be combined to compute an approximation of the Gromov-Hausdorff distance of metric spaces with thousands of points. Algorithm 3.8 computes extreme sets which comprise many of the relevant features of the metric spaces if their cardinalities are chosen suitably. We compute a maximal relation between these extreme sets with Algorithm 3.1 for a given estimate for the minimal distortion δ. This step also ensures that we can construct a good correspondence if the extreme sets of both metric spaces do not overlap completely, i. e., only a subset of the extreme set in both spaces is approximately matched by an optimal correspondence. The obtained relation is then extended by Algorithm 3.6.

Applying this framework to some shapes included in the TOSCA project [BBK08] on a standard laptop[2], we notice that the results are satisfactory and well resemble the expected results when matching different poses of a shape, see Figure 3.21. Note, however, that the right leg and arm of the shape on the left in Figure 3.21a are matched to the left leg and arm of the shape on the right, respectively. This is due to the fact that the distortion is invariant under isometric transformations of the spaces, i. e., self-symmetries of the shapes cannot be distinguished by our approach. Detecting and classifying (approximate) full and

[2]Intel Dual Core i5 CPU at 2.5 GHz and 8 GB of RAM; algorithms implemented in Matlab



(a) David



(b) Wolf

Figure 3.21: Relations between extreme sets of shapes in different poses. Green and red points represent matched and unmatched points of the extreme set. The green lines illustrate assignments between extreme points, while the blue lines depict assignments inducing the maximal distortion of the relation.

Input: Metric spaces $(X, d_X), (Y, d_Y)$, size of extreme sets s_e, size of reference set s_r, estimate for Gromov-Hausdorff distance $\delta \geq 0$
Output: Correspondence $R \in \mathcal{R}(X, Y)$, relation $R_B \subseteq B_X \times B_Y$

1 $B_X \leftarrow$ `extremeSet`$((X, d_X), s_e)$
2 $B_Y \leftarrow$ `extremeSet`$((Y, d_Y), s_e)$
3 $R_B \leftarrow$ `computeSGHD`$((B_X, d_X), (B_Y, d_Y), \delta)$
4 $R_E \leftarrow$ `extendCorrespondenceRef`$((X, d_X), (Y, d_Y), R_B, s_r)$
5 $R \leftarrow$ `improveCorrespondence`$((X, d_X), (Y, d_Y), R_E)$

Algorithm 3.8: computeCorrespondence

partial symmetries of shapes is an active field of research on its own, see [Mit+13] for a recent survey.

Numeric results for several combinations of input parameters s_e and δ (cf. Table 3.1) are shown in Table 3.2. Since the meshes from the TOSCA project [BBK08] which represent different poses of the same shape are obtained by modifying an original mesh, there is a canonical correspondence $R_C := \{(x_i, y_i) \mid i \in \{1, \ldots, n\}\}$, where $X := \{x_1, \ldots, x_n\}$ and $Y := \{y_1, \ldots, y_n\}$. It constitutes a good estimate for an optimal correspondence. Note that Algorithm 3.8 does not use any information on the sorting of the sets and therefore does not benefit from this special setting, but is able to find a better correspondence for several combinations of parameters.

Input	s_e	Cardinality of extreme sets
	δ	Bound on the distortion of R_B
Output	R_B	Relation between extreme sets
	R	Correspondence between X and Y

Table 3.1: Notation used in Algorithm 3.8 and in the following tables.

The time needed for each run depends mostly on the size of the metric spaces; the parameters s_e and δ do not have a big impact. This is due to the fact that the computational cost for the computation of an optimal relation between the extreme sets in both shapes is usually less than 10ms, and therefore negligible, for the range of parameters supplied here. The running time is dominated by the extension of the correspondence, which depends almost solely on the size of the metric spaces. On the other hand, the distortion of the computed correspondence depends heavily on the input parameters. The quality of the relation R_B between the extreme sets has a major impact on the final solution, and is rather unstable with respect to these parameters.

When matching similar — but not identical — shapes, the computational results are still well within realistic boundaries, see Figure 3.22 and Table 3.3. Although in this case there is no canonical correspondence, we can use the difference between diameters of shapes as a trivial lower bound for the distortion of a correspondence. For the results presented in Table 3.3, this yields an a posteriori bound on the approximation ratio of $\frac{1/2\Gamma(R)}{d_{GH}(X,Y)} \leq \frac{\Gamma(R)}{|\mathrm{diam}(X) - \mathrm{diam}(Y)|} \leq 3/2$ for the results produced by Algorithm 3.8 for a suitable choice of parameters.

However, the results shown in Figure 3.22 suggest that the Gromov-Hausdorff distance is not necessarily the most suitable measure to compute visually satisfying correspondences.

s_e	δ	$\lvert R_B\rvert$	$\Gamma(R)$	time
6	10.0	6	18.62344	122.7 s
6	12.0	6	28.02216	132.2 s
6	14.0	6	20.22373	127.2 s
9	10.0	7	24.59136	131.8 s
9	12.0	7	14.88461	123.9 s
9	14.0	7	14.88461	124.8 s
12	10.0	11	21.15703	125.2 s
12	12.0	11	21.15703	126.0 s
12	14.0	11	21.68575	125.3 s
15	10.0	12	15.97880	130.6 s
15	12.0	13	16.31404	141.0 s
15	14.0	14	17.12237	122.5 s

(a) (X, d_X) and (Y, d_Y) derived from shapes depicted in Figure 3.21a with 5000 points, diameters $\operatorname{diam}(X) = 206.01$, $\operatorname{diam}(Y) = 206.68$ and $\Gamma(R_C) = 18.03$ for the canonical correspondence R_C.

s_e	δ	$\lvert R_B\rvert$	$\Gamma(R)$	time
6	10.0	6	16.49248	89.4 s
6	12.0	6	19.14281	87.2 s
6	14.0	6	19.14281	87.7 s
9	10.0	7	43.63896	100.1 s
9	12.0	8	30.77307	104.7 s
9	14.0	8	18.88477	95.9 s
12	10.0	9	31.71154	100.6 s
12	12.0	10	18.57190	98.9 s
12	14.0	10	18.12550	124.7 s
15	10.0	11	15.03307	95.7 s
15	12.0	13	15.98435	96.2 s
15	14.0	13	15.60199	130.1 s

(b) (X, d_X) and (Y, d_Y) derived from shapes depicted in Figure 3.21b with 4344 points, diameters $\operatorname{diam}(X) = 179.91$, $\operatorname{diam}(Y) = 178.99$ and $\Gamma(R_C) = 15.67$ for the canonical correspondence R_C.

Table 3.2: Results of Algorithm 3.8 for a choice of parameters (cf. Table 3.1) for the metric spaces $(X, d_X), (Y, d_Y)$ derived from 2-dimensional manifolds representing a shape in different poses, see Figure 3.21.

s_e	δ	$\lvert R_B\rvert$	$\Gamma(R)$	time
6	15.0	4	114.60709	114.9 s
6	20.0	4	114.60709	112.3 s
6	25.0	4	34.27743	104.5 s
9	15.0	4	98.07209	106.8 s
9	20.0	4	34.27743	106.9 s
9	25.0	5	34.27743	107.4 s
12	15.0	6	102.07595	113.5 s
12	20.0	6	35.27883	105.3 s
12	25.0	7	46.68273	106.3 s
15	15.0	8	35.97512	104.1 s
15	20.0	8	34.73239	112.7 s
15	25.0	9	46.68273	106.4 s

(a) (X, d_X) and (Y, d_Y) derived from shapes depicted in Figure 3.22a with $|X| = |Y| = 5000$ and $\operatorname{diam}(X) = 206.68$, $\operatorname{diam}(Y) = 230.50$.

s_e	δ	$\lvert R_B\rvert$	$\Gamma(R)$	time
6	15.0	2	57.15253	141.9 s
6	20.0	3	67.16962	166.0 s
6	25.0	4	42.27561	150.3 s
9	15.0	4	70.92869	169.6 s
9	20.0	5	69.79849	135.7 s
9	25.0	6	70.40876	147.8 s
12	15.0	6	45.35137	143.3 s
12	20.0	7	50.15459	150.8 s
12	25.0	8	70.65079	154.8 s
15	15.0	8	68.54642	159.0 s
15	20.0	10	67.88702	150.0 s
15	25.0	10	65.09951	145.5 s

(b) (X, d_X) and (Y, d_Y) derived from shapes depicted in Figure 3.22c with $|X| = |Y| = 5000$ and $\operatorname{diam}(X) = 199.62$, $\operatorname{diam}(Y) = 172.00$.

Table 3.3: Influence of base size and estimate δ on Algorithm 3.8 for metric spaces $(X, d_X), (Y, d_Y)$ derived from 2-dimensional manifolds representing similar shapes, cf. Figure 3.22.

While the correspondence shown in Figures 3.22a and 3.22b between a male and a female person and a wolf and dog shape are as one would expect, the results shown in Figure 3.22c reveal that this is not necessarily true for all shape pairings. Although this relation can be extended to a near optimal correspondence (cf. Table 3.3b), the dogs right hind leg is matched to the cats ear, while the dogs snout is matched to the cats tail. In case of Figure 3.22c it is easy to see that, since all extremities have approximately the same length, these parts are indistinguishable as metric spaces once a certain margin of error is accepted. In this case, the distortion is induced by the assignment of the dogs front legs to the cats hind legs, thus the assignment of the dogs hind legs and tail is almost arbitrary.

It is also interesting to note that the framework described above performs very favourably for random metric spaces, given that an estimate of the Gromov-Hausdorff distance is known. We tested the framework on metric spaces generated by first computing an undirected random graph $G_1 \coloneqq (V_1, E_1, w_1)$ with probability γ for the existence of each edge and with uniformly distributed edge weights. A second graph $G_2 \coloneqq (V_1, E_1, w_2)$ was derived from this graph by adding a uniformly distributed random number to each edge weight

$$w_2(e) \coloneqq \max\{0, w_1(e) + r\} \text{ for } r \in [-\alpha/2, \alpha/2] \text{ uniformly distributed}$$

with the disturbance parameter $\alpha \geq 0$. We then computed the metric completion of these graphs, i. e., the distance matrix consists of the length of shortest path between all vertices. The results can be seen in Table 3.4.

s_e	δ	$\lvert R_B \rvert$	$\Gamma(R)$	time
6	5.0	6	23.58297	0.1 s
6	10.0	6	27.65317	0.1 s
6	15.0	6	24.67791	0.2 s
9	5.0	8	26.00985	0.2 s
9	10.0	9	27.97525	0.2 s
9	15.0	9	25.43381	0.2 s
12	5.0	9	31.61614	0.3 s
12	10.0	12	30.93351	0.3 s
12	15.0	12	28.51344	0.3 s
15	5.0	11	28.70478	0.5 s
15	10.0	15	26.63995	0.5 s
15	15.0	15	25.38538	1.5 s

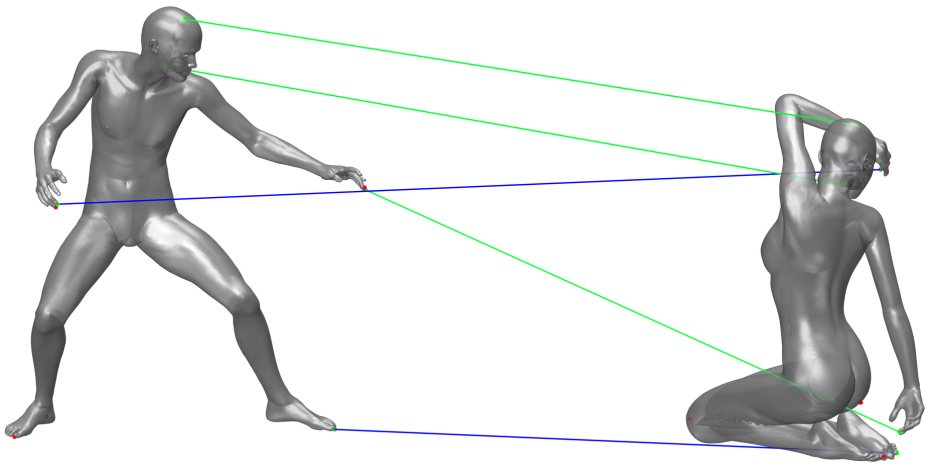
(a) $|X| = |Y| = 250$ with diameter $\operatorname{diam}(X) = 64.92$, $\operatorname{diam}(Y) = 65.16$ and $\Gamma(R_C) = 24.62$ for the canonical correspondence R_C.

s_e	δ	$\lvert R_B \rvert$	$\Gamma(R)$	time
6	5.0	6	30.47338	0.2 s
6	10.0	6	33.50918	0.2 s
6	15.0	6	31.65600	0.2 s
9	5.0	9	40.02511	0.2 s
9	10.0	9	31.03226	0.3 s
9	15.0	9	36.39717	0.3 s
12	5.0	11	31.84973	0.4 s
12	10.0	11	32.65692	0.4 s
12	15.0	12	33.66973	0.4 s
15	5.0	11	25.15262	0.5 s
15	10.0	15	37.25498	0.6 s
15	15.0	15	35.33621	5.4 s

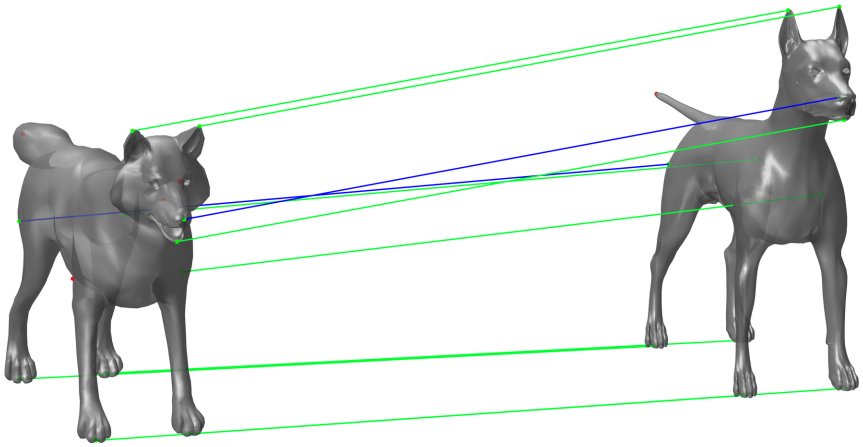
(b) $|X| = |Y| = 300$ with diameter $\operatorname{diam}(X) = 87.33$, $\operatorname{diam}(Y) = 87.43$ and $\Gamma(R_C) = 28.20$ for the canonical correspondence R_C.

Table 3.4: Influence of base size and bound on Algorithm 3.8 for random metric spaces $(X, d_X), (Y, d_Y)$.

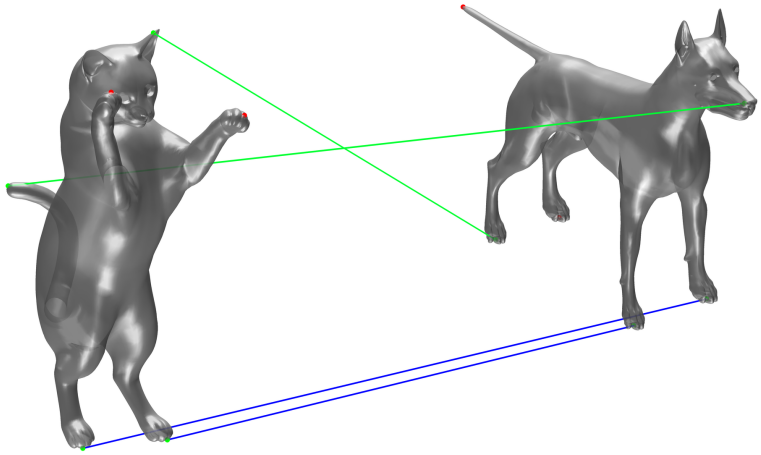
If the disturbance α is small, the canonical correspondence $R_C \coloneqq \{(v, v) \mid v \in V\}$ is again a reasonable estimate for an optimal correspondence, but Algorithm 3.8 finds a better correspondence in both instances for several combinations of parameters. This is in accordance with the results of various other tests performed on such random metric spaces, which, for the sake a compact presentation, are not shown here. This suggests that the concepts of bases and



(a) David and Victoria



(b) Wolf and dog



(c) Cat and dog

Figure 3.22: Relations between extreme sets of different shapes.

extreme points are not only suitable for highly structured data sets like the ones obtained from 2-manifolds with few distinct attributes like extremities. Even large random metric spaces tend to contain small characteristic subspaces. The last row in Tables 3.4a and 3.4b discloses the impact of the running time of the exact solution of Problem δ-p-SGHD on the total running time of algorithm. Once a certain size of the input for this problem is exceeded, the running time increases drastically and, in this case, dominates the computational cost of extending the correspondence.

Furthermore, a correspondence computed with this framework can be used as an estimate for an optimal solution for the p-Gromov-Hausdorff distance problem for $p \in \mathbb{N}$. In the cases where a canonical correspondence is known, the results are comparable, and sometimes slightly better, than the canonical correspondence. An interesting topic for further research is whether results similar to Theorems 3.43 and 3.47 can be achieved for a greedy (or other suitable) extension of a relation.

3.4 Partial Matching

Similar to the approach in Chapter 2, we can also consider non-rigid shape matching as a multi-criteria optimization problem. As in the previous chapter, suitable criteria for the quality of a relation $R \subseteq X \times Y$ are the distortion of the relation $\Gamma(R)$ and the number of points of X and Y covered by R.

While the algorithm presented in the proof of Theorem 3.47 is very similar to the algorithm for aligning point clouds given in Chapter 2, it cannot be as easily modified to obtain an approximation of the Pareto front. The main problem arising is that, rather counterintuitively, subspaces of a metric space can have a more complex structure than the original space. Consider Figure 3.23, in which a metric space with 1-base $\{x_4\}$ of cardinality 1 is depicted. In the subspace obtained by removing x_4, one needs 2 points to construct a 1-base.

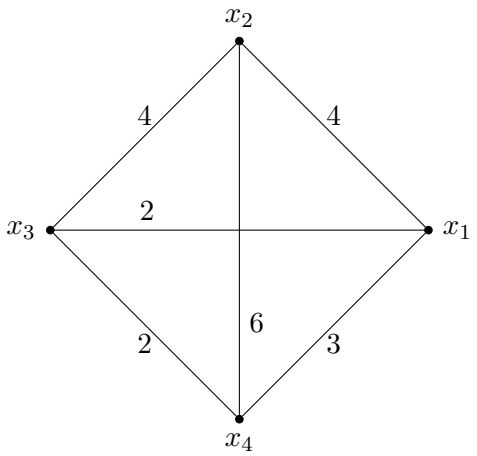


Figure 3.23: The property of having small ε-bases is not necessarily preserved in subspaces of a metric space: x_4 is a 1-base of the depicted metric space, but there exists no 1-base of cardinality 1 in the subspace induced by the points $\{x_1, x_2, x_3\}$.

In contrast to the situation in Section 2.4, we cannot bound the cardinality of ε-bases of subspaces we would need to consider. Consequently, it is not sufficient to evaluate all subsets

of both metric spaces of cardinality bounded by the size of ε-bases of the metric spaces. As pointed out in the counterexample given in Figure 3.23, these subsets may yield solely ε'-bases with $\varepsilon' > \varepsilon$.

The question whether at least a bound on ε' can be given in terms of ε remains open:

Open question *Let (X, d_X) be a metric space which contains an ε-base of cardinality $N \in \mathbb{N}$ for $\varepsilon > 0$. Is there a constant $r \in \mathbb{R}$, such that each induced subspace (X', d_X), $X' \subseteq X$ contains an $r \cdot \varepsilon$-base of cardinality N?*

Alternatively, a bound on the minimum cardinality of ε-bases of a subspace in terms of the cardinality of an ε-base would be of interest.

Open question *Let (X, d_X) be a metric space which contains and ε-base of cardinality $N \in \mathbb{N}$ for $\varepsilon > 0$. Is there a constant $r \in \mathbb{R}$, such that each induced subspace (X', d_X), $X' \subseteq X$ contains an ε-base of cardinality $r \cdot N$?*

In the positive case, either of these bounds would enable a Pareto approximation of the problem by a slight modification of the algorithm given in Theorem 3.47. As in Algorithm 2.3, the error induced by including a point in the extension could be estimated and thus an approximation of the Pareto front could be achieved.

Still, the machinery developed in the previous section for practical examples can be easily adapted to the problem of partially matching metric spaces. Reconsidering Algorithm 3.8, we already used a partial matching approach to account for the fact that extreme sets are in general not unique and a chosen extreme set might contain points which have no suitable matching partner in the other extreme set. By modifying the greedy extension to only include points which have a suitable matching partner in the other space, we can greedily extend a relation, see Algorithm 3.9. Algorithms 3.5 and 3.6 can be modified accordingly. Results obtained with this slightly modified framework are presented in Figure 3.24 and Table 3.5.

s_e	δ	$\lvert R_B \rvert$	$\lvert \mathcal{P}_1(R) \rvert$	$\lvert \mathcal{P}_2(R) \rvert$	$\Gamma(R)$	time
6	35.0	3	4087	2383	49.43997	96.3 s
6	40.0	3	4197	2533	47.55256	104.1 s
6	45.0	3	5000	3534	50.87202	125.5 s
9	35.0	5	5000	3442	42.37676	125.2 s
9	40.0	5	5000	3581	48.16567	139.3 s
9	45.0	5	5000	3728	51.57740	149.8 s
12	35.0	8	5000	3448	43.42775	126.7 s
12	40.0	8	5000	3583	47.84064	140.3 s
12	45.0	8	5000	3752	51.71773	143.4 s

Table 3.5: Partial matching of metric spaces (X, d_X), (Y, d_Y) with $|X| = |Y| = 5000$ points and $\operatorname{diam}(X) = 225.68, \operatorname{diam}(Y) = 279.69$.

The algorithm successfully detects the human part of the centaur and matches it to the person. Table 3.5 shows that almost all relations computed cover the metric space (X, d_X) obtained from the mesh representing the person completely, cf. column $|\mathcal{P}_1(R)|$. The metric space derived from the centaur is covered only partially, since not all parts can be matched to the person with acceptable error. This can be seen in column $|\mathcal{P}_2(R)|$ of Table 3.5.

Input: Metric spaces $(X, d_X), (Y, d_Y)$, correspondence $R_B \in \mathcal{R}(B_X, B_Y)$, estimate for the distortion $\delta \geq 0$
Output: Relation $R \in \mathcal{R}(X, Y)$

$R \leftarrow R_B$
$X' \leftarrow X \setminus \mathcal{P}_1(R)$
for $x \in X'$ **do**
  $c \leftarrow \min_{y \in Y} \max_{(x',y') \in R_B} |d_Y(y', y) - d_X(x', x)|$
  if $c \leq \delta$ **then**
    $y^* \leftarrow \operatorname{argmin}_1 \max_{(x',y') \in R_B} |d_Y(y', y) - d_X(x', x)|$
    $R \leftarrow R \cup \{(x, y^*)\}$
    $\zeta \leftarrow \max\{c, \zeta\}$
  end
end
$Y' \leftarrow Y \setminus \mathcal{P}_2(R)$
for $y \in Y'$ **do**
  $c \leftarrow \min_{x \in x} \max_{(x',y') \in R_B} |d_Y(y', y) - d_X(x', x)|$
  if $c \leq \delta$ **then**
    $x^* \leftarrow \operatorname{argmin}_1 \max_{(x',y') \in R_B} |d_Y(y', y) - d_X(x', x)|$
    $R \leftarrow R \cup \{(x^*, y)\}$
    $\zeta \leftarrow \max\{c, \zeta\}$
  end
end

Algorithm 3.9: extendRelation

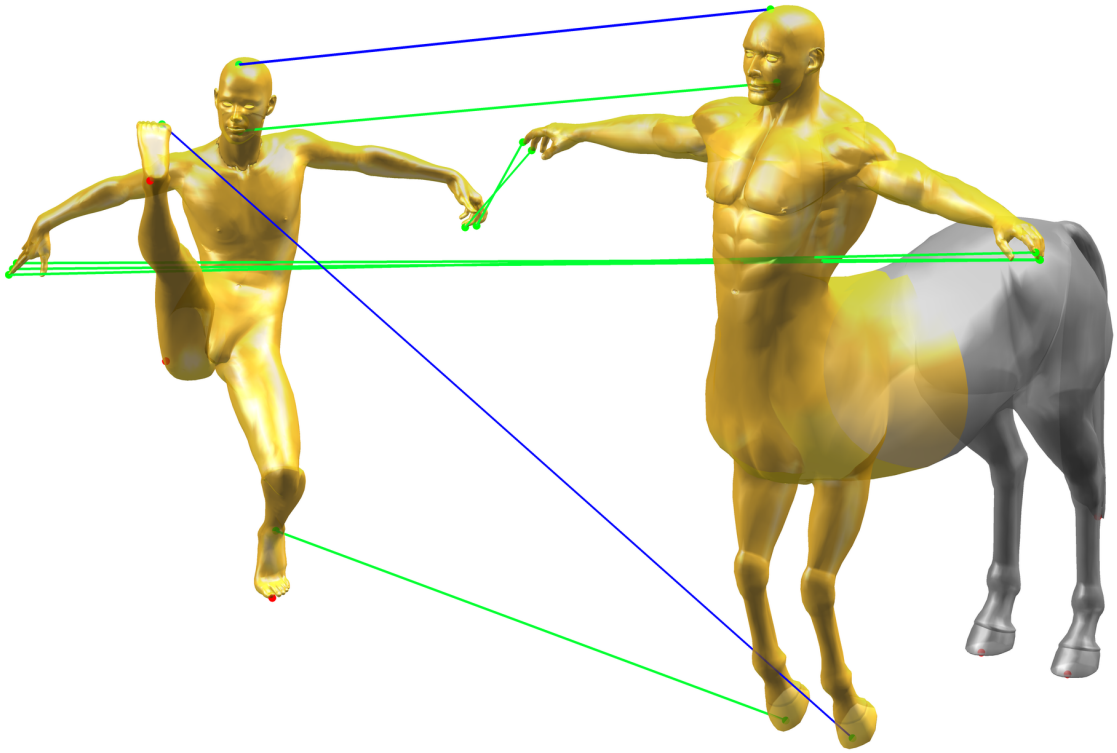


Figure 3.24: Relation between different shapes computed with Algorithm 3.8. The parts covered by the relation are highlighted in yellow.

Chapter 4

Dense k-Subgraph Problem

The problem of identifying a subgraph with maximal weight subject to some sort of size constraint in a given graph is a well-known and widely studied problem in operations research. Various interpretations of this problem setting have been investigated in the literature, with applications in such diverse fields as facility location [Pis06] and (social) network analysis [AC09]. Furthermore, non-rigid shape matching can also be reduced to this problem. The bottleneck p-Gromov-Hausdorff distance of two metric spaces (X, d_X), (Y, d_Y) can be expressed as the maximal weight of a subgraph with cardinality $|X| = |Y|$ in a weighted graph modelling the error induced by pairs of matched points, cf. Section 3.3.2.

The corresponding optimization problem, which asks for the densest subgraph with exactly k vertices of a weighted or unweighted graph is known as dense k-subgraph problem. It is a natural generalization of the maximum clique problem (readily implying $\mathcal{NP}$-hardness), and has therefore been subject to a substantial amount of research.

Problem 4.1 (Dense k-subgraph (DkS))
Input: Weighted graph $G \coloneqq (V, E, w)$, $k \geq 2$, $\mathbb{N} \geq 0$.
Question: Is there a set of vertices $S \subseteq V$ such that $|S| = k$ and $\sum_{e \in E(S)} w(e) \geq N$.

Variations of this particular problem, asking for the densest subgraph with at least (Dal$k$S) or at most (Dam$k$S) k vertices have also been considered.

In the context of location problems, the weighted DkS problem is also often referred to as maximal dispersion (or 1-dispersion) problem. As these problems usually arise from weighted graphs representing distances in an Euclidean space, the weights are often assumed to satisfy the triangle inequality. In addition to these applications, our interest in the DkS problem is motivated by the problem of computing the bottleneck p-Gromov-Hausdorff distance and a real-world application in forestry, for which this is not the case.

Not surprisingly, both heuristics and exact solution methods for the DkS problem benefit significantly from smaller instances. As in the previous chapters, we investigate methods which decrease the size of an instance significantly, while retaining an approximate solution. The former methods try to identify few points which describe an instance accurately. To the contrary, in this chapter, which is based on joint work with Steffen Borgwardt [BS14], we present an approach which identifies and removes vertices which cannot contribute significantly to a solution. The resulting graph retains a $(1 + \frac{1}{k-1})$-approximation[1] for the DkS problem on graphs that do not need to satisfy the triangle inequality.

The key result we use is that we can efficiently determine a so-called Δ-*core* of a graph G, i. e., an induced subgraph of G in which each vertex has weighted degree no less than Δ, that still contains such an approximate solution for a suitable threshold Δ. The threshold is

[1] In [BS14], the approximation factor is erroneously stated as $(1 + \frac{1}{k})$

dependent on the edge weight contribution of a vertex in the selection of a k-vertex subset of the graph. Vertices whose possible contribution lies below the threshold are not vital for an approximate solution for the DkS problem and can thus be dropped from the input graph without deteriorating the objective value significantly.

If the resulting Δ-core is small enough, this enables us to find an exact solution (or approximate solution with low error) to the problem on this Δ-core. We give an algorithm which computes such a subgraph in polynomial time. Thus, the computational cost of finding an approximate solution for the DkS problem in this way depends almost solely on the number of vertices in the remaining Δ-core. Due to this, it is important to find a large threshold Δ for the graph reduction, since this allows for the pruning of more vertices. For this purpose, we devise an efficient algorithm that computes a provably optimal threshold.

Unfortunately, the Δ-cores arising from the reformulation of Problem p-BGHD are usually still rather large. The edge weights in these graphs are relatively evenly distributed, which causes a low threshold. In combination with the huge size of the graphs obtained from the construction in Section 3.3.2, this typically implies that not enough vertices can be pruned to admit an exact solution of the problem on the resulting graphs with current methods.

However, we present an application in forestry which benefits considerably from the methods introduced in this chapter, often allowing $90 - 95\%$ of all vertices of the graph to be removed. At first, we briefly survey the relevant literature.

From a theoretical point of view, it is known that the DkS problem is $\mathcal{NP}$-hard even for bipartite graphs of maximal degree 3 [FS97], comparability graphs, chordal graphs [CP84] and planar graphs [KB91]. Khot [Kho06] shows that there can be no PTAS for the dense k-subgraph problem under the standard complexity theoretic assumption $\mathcal{NP} \not\subseteq \bigcap_{\varepsilon>0} \mathcal{BPP}(2^{n^\varepsilon})$ (see [AB09, Definition 7.3] for a definition of $\mathcal{BPP}$), but no stronger bound on the inapproximability is known. In particular, it is unknown whether the problem is in $\mathcal{APX}$ or not.

Several approaches for solving the DkS problem exactly have been investigated. Macambira and Souza [MS00] and Sørensen [Sø04] consider a mixed integer programming formulation for the problem of finding a k-clique of maximal weight in a vertex- and edge-weighted complete graph. They derive several facets of the underlying k-clique polytope, which are incorporated in a branch-and-cut algorithm. Hunting, Faigle, and Kern [HFK01] propose an approach using cutting planes in a Lagrangian relaxation of the integer linear programming formulation, while [BEP09] describes a tight convex reformulation of a quadratic model for the problem. Due to the difficulty of the problem, exact solutions are in general only possible for small instances. Macambira and Souza [MS00] and Hunting, Faigle, and Kern [HFK01] consider weighted complete graphs with up to 48 vertices and Sørensen [Sø04] reports results for graphs with at most 61 vertices and $k = \lfloor n/2 \rfloor$. Billionnet, Elloumi, and Plateau [BEP09] consider instances with up to 100 vertices with varying densities and $k = \lfloor i\frac{n}{4} \rfloor$ for $i \in \{1, 2, 3\}$.

For larger instances, an exact solution is computationally too expensive in most cases. Several polynomial time approximation algorithms are considered in the literature. Asahiro et al. [Asa+00] analyse a greedy algorithm for the maximal weight k-clique problem which realizes an approximation ratio of $\mathcal{O}(\frac{n}{k})$. Feige and Langberg [FL01] present an algorithm based on an SDP relaxation which achieves an approximation ratio of $\frac{n}{k}$, which is further improved in [HYZ02], for a range of values of $\frac{n}{k}$.

Kortsarz and Peleg [KP93] present an algorithm with approximation ratio $\mathcal{O}(n^\alpha)$ for a constant $\alpha > 0$ for the unweighted DkS problem, by combining several algorithms which achieve good results on specific instances. The algorithms and their analysis are refined in

[FPK01] and [Gol09]. The currently best known approximation ratio independent of k is achieved by Bhaskara et al. [Bha+10] in the form of $n^{1/4+\epsilon}$. In [FPK01], it is also mentioned that an approximation algorithm for the unweighted DkS problem can be generalized to the weighted version, incurring an additional factor of $\log(n)$ for the approximation ratio.

Constant factor approximations are known for slight variations of the DkS problem: Goldstein [Gol09] proposes a 2-approximation for the problem of finding the densest subgraph with at least k vertices (Dal$k$S) by transforming a relaxed version into a submodular function minimization problem, which can be solved in polynomial time. This result also holds for the weighted version with non-negative edge weights.

Hassin, Rubinstein, and Tamir [HRT97] construct a 2-approximation algorithm for the problem of finding l disjoint induced subgraphs, each of size k, of a weighted graph satisfying the triangle inequality, such that the sum of their weights is maximized. In [LMZ08], a 3-approximation algorithm for the DkS problem on chordal graphs is introduced. Furthermore, the problem without size constraint can be transformed to a flow problem and is therefore solvable in polynomial time [Gol84].

As in the previous chapters, the size of the instances arising in our applications is rather large. Thus, established methods do not yield satisfactory results in reasonable running time. We present an approach which is motivated by the special properties of the application in forestry. Before we describe this application and its specific properties in detail, we introduce the following basic graph theoretic notation in addition to that defined in Section 1.3. In a weighted graph $G \coloneqq (V, E, w)$, $w(E')$ denotes the sum of weights of the edges in E' and $w(V')$ the sum of weights of edges in $E(V')$. The *potential* or *weighted degree* of a vertex v with respect to a subgraph G' of G is defined as the sum of the weights of adjacent edges:

$$p_{G'}(v) \coloneqq \sum_{v' \in N_{G'}(v)} w(\{v, v'\}).$$

If the subgraph G' is clear from the context, we also use the notation $p(v)$. For $v_i \in V$, we also write $p(i)$ if it allows for a simpler notation. Let us now describe the application in forestry.

4.1 An Application in Forestry

In many forestry regions in Northern Bavaria, an efficient cultivation has become impossible due to heritage law and frequent change of ownership. The average size of a lot is less than 0.3 hectares. Further, the lots themselves often have a shape badly suited for cultivation (e. g., several dozens of meters long, but only a few meters wide). Finally, even in small regions, there are typically several hundred different owners of very small lots. Figure 4.1 depicts an example[2] consisting of 560 lots that belong to 227 owners.

With so many different owners, a classical land consolidation approach is difficult to implement in practice. Instead, one initiates what is called a voluntary land exchange. In such a process, the structural partition of the forestry region into lots remains unchanged, but the ownership of the existing lots may be redistributed combinatorially. This approach is similar to the exchange of lend-lease agreements in agricultural regions, see [BG04; BBG11].

[2]The figure was created using the licensed software "Arborchange". To respect data privacy, the figure does not depict the ownership structure of a real-world region, but instead an example specifically created for visualization purposes. It shares all of the common and representative properties of the real-world regions investigated.

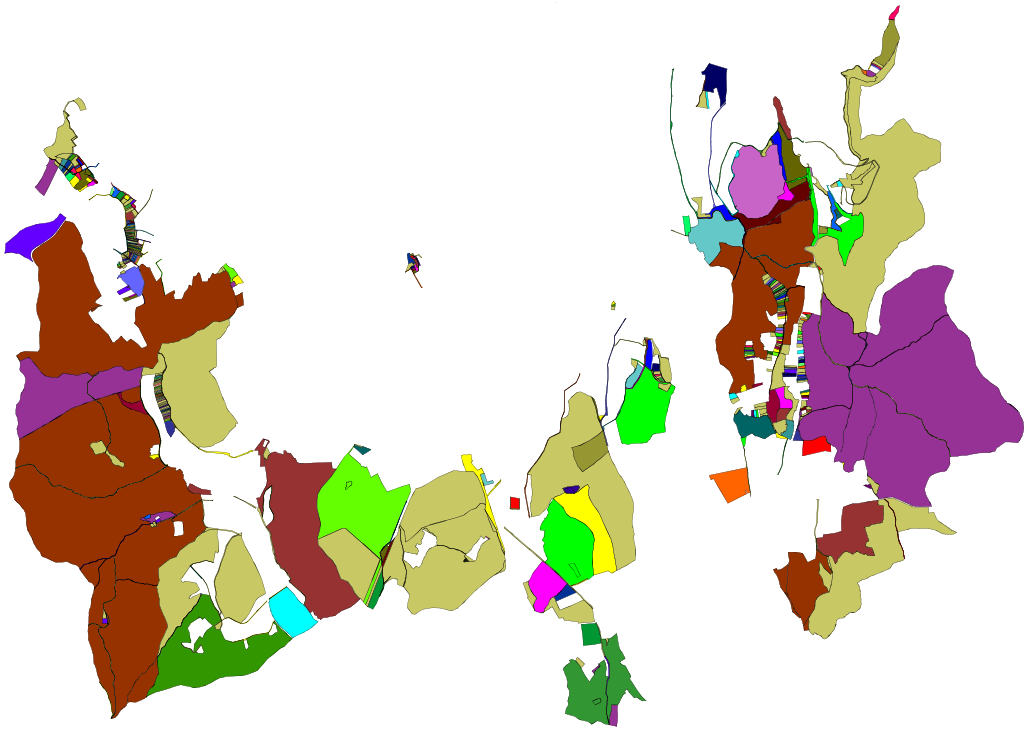


Figure 4.1: A typical forestry region. Different colours represent different owners.

To improve the cost-effective structure of the region by means of such a redistribution, larger connected areas of lots that belong to a single owner should be created, i. e., adjacent lots should be assigned to the same owner. Of course, the total value of the lots of the owners has to remain about the same in the redistribution process.

Due to the high number of owners even in small forestry regions, it is not possible to contact or negotiate with all owners in practice. Thus, one selects a small number of k owners that are asked to participate on a voluntary basis. The criterion by which this choice is made is very important. A natural approach is to try to find the k owners which provide maximal room for improvement of the cost-effective structure in the region.

As mentioned above, the most significant measure for the improvement of the cost-effective structure is the creation of larger areas of lots that belong to the same owner during such a redistribution process. An intuitive way to judge how much potential an owner contributes to a redistribution process is the total length of boundary between his own lots and other owners' lots. For example, consider two owners in a forestry region. One of them has only a single, huge area of connected lots, and thus only shares a rather low total length of boundary with other owners. Conversely, the second one owns many small lots that, in total, have the same size as the other owner's area of lots. The fragmented nature of these small lots typically means that the boundary to other lots is much longer. In this case, the second owner would be the better choice for participation in the land exchange process.

By assigning adjacent lots to the same owner, one might be able to get rid of their common boundary. Further, large boundary lengths are often connected to unfavourable lot shapes: a very long lot that is only a few meters wide may have a small area, but nonetheless the boundary between it and another lot that belongs to a different owner can be very large. By

preferring owners whose lots have large total boundary lengths, one also implicitly tries to improve on these bad lot shapes.

The underlying mathematical problem can be modelled as follows. Let $G \coloneqq (V, E, w)$ be an undirected weighted graph. Its vertices $V \coloneqq \{v_1, \ldots, v_n\}$ correspond to the n owners in the region. Further, there is an edge $\{v_i, v_j\} \in E$ if and only if the owners v_i and v_j have at least one common boundary between their lots. The weight function $w\colon E \longrightarrow \mathbb{R}$ denotes the total length of all boundaries in between the respective owners' lots.

As shown in our empirical studies, the graphs constructed in this way share some special properties. On the one hand, the number of edges is typically bounded by $10n$, and there are vast differences in the edge weights and potential of their vertices. On the other hand, the edge weights do not satisfy the triangle inequality.

In the following, we present algorithms which exploit the special properties of these graphs.

4.2 Threshold-based Pruning and Approximate Solution

The threshold of an ordered set of vertices in a graph takes a central role, as we can use it to identify vertices which cannot contribute a lot of edge weight to a subgraph induced by k vertices. Such vertices are not vital for a good solution of the DkS problem.

Definition 4.2 (Threshold)
Let $\mathcal{S} \coloneqq (v_1, \ldots, v_k)$ be an ordered set of k vertices in a weighted graph $G \coloneqq (V, E, w)$, $V_i \coloneqq \{v_1, \ldots, v_i\}$ for $i \in \{1, \ldots, k\}$ and $V_0 \coloneqq \emptyset$. The threshold $\Delta \coloneqq \Delta(\mathcal{S})$ of this ordered set is defined as

$$\Delta(\mathcal{S}) \coloneqq \min_{i=1,\ldots,k-1} \delta^i,$$

where

$$\delta^i \coloneqq \max\Big\{ w(V_i) - w(V_{i-1}), \frac{1}{2}\Big(w(V_{i+1}) - w(V_{i+1} \setminus \{v_i\})\Big)\Big\}.$$

If the ordered set $\mathcal{S}$ is clear from the context, we use the shorter notation Δ for the threshold. When we have to distinguish several ordered sets, e. g., $\mathcal{S}_1, \mathcal{S}_2$ or $\mathcal{S}_o, \mathcal{S}_g$, we use the notation δ_1^i, δ_2^i and δ_o^i, δ_g^i to refer to the corresponding values of δ^i.

The threshold Δ is a lower bound for the weight increase by sequentially adding vertices v_i, $i \in \{1, \ldots, k\}$ to V_{i-1}. Note that the threshold depends on the order of the set and that the value for the last vertex is ignored, in the sense that there is no computation of a value δ^k. For each vertex v_i, we consider two values: The difference $w(V_i) - w(V_{i-1})$ represents the additional edge-weight gained by adding vertex v_i to V_{i-1}. The second term $\frac{1}{2}(w(V_{i+1}) - w(V_{i+1} \setminus \{v_i\}))$ expresses the weight lost by removing v_i from V_{i+1}, which is a lower bound for (one-half) the contribution of v_i and v_{i+1} combined. This ensures that the vertices of an ordered set with a large threshold Δ do not need to form a connected component. The vertex v_i may be disconnected from the preceding vertices (or have very low potential towards them), and this may be compensated by a vertex v_{i+1} when the addition of both v_i and v_{i+1} contributes at least 2Δ.

An example is given in Figure 4.2. The ordered set $\mathcal{S} \coloneqq (v_1, v_2, v_3, v_4)$ has threshold $1/2$: Adding vertex v_3 to (v_1, v_2) yields no gain in edge weight, but v_3, v_4 combined increase the weight by 1. The per-vertex gain is thus $1/2$ for these two vertices. The threshold of the permutation (v_1, v_3, v_4, v_2) of the set is 0, since neither v_1 nor (v_1, v_3) induce a positive weight.

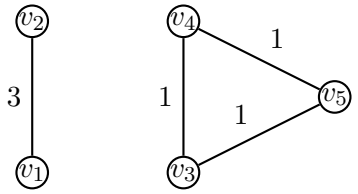


Figure 4.2: The threshold of the ordered set $\mathcal{S} = (v_1, v_2, v_3, v_4)$ is given by $\Delta(\mathcal{S}) = \min\{3/2, 3, 1/2, 1\}$, while $\Delta((v_1, v_3, v_4, v_2)) = \min\{0, 1/2, 1, 3\}$.

The bound on the approximation ratio of our approach greatly depends on the fact that removing a subset B of m vertices from an ordered set $\mathcal{S}$ with threshold Δ implies a loss of edge weight of at least $(m-1)\Delta$. In other words, adding the vertices in B to $A := \mathcal{S} \setminus B$ yields a gain of weight of at least $(m-1)\Delta$.

Lemma 4.3
Let $\mathcal{S} \coloneqq (v_1, \ldots, v_k)$ be an ordered set of vertices, Δ the corresponding threshold, and let $A \subseteq \mathcal{S}$. Then we have

$$w(\mathcal{S}) \geq w(A) + (|\mathcal{S} \setminus A| - 1)\Delta.$$

Proof. Let $V_i \coloneqq \{v_1, \ldots, v_i\}$, $t \coloneqq |\mathcal{S} \setminus A|$ and $j_1 < \ldots < j_t$ such that $(v_{j_1}, \ldots, v_{j_t}) = \mathcal{S} \setminus A$. Consider $i \in \{1, \ldots, t-1\}$. By definition of the threshold, we have either $w(V_{j_i}) - w(V_{j_i-1}) \geq \Delta$ or $w(V_{j_i+1}) - w(V_{j_i+1} \setminus \{v_{j_i}\}) \geq 2\Delta$. In the first case, adding vertex v_{j_i} increases the weight of $A \cup \{v_{j_1}, \ldots, v_{j_{i-1}}\}$ by at least Δ. In the second case, if $v_{j_i+1} \in A$, the weight of $A \cup \{v_{j_1}, \ldots, v_{j_{i-1}}\}$ is augmented by at least 2Δ by adding v_{j_i}. If $v_{j_i+1} \notin A$, we have $j_i + 1 = j_{i+1}$ and therefore adding vertices v_{j_i} and $v_{j_{i+1}}$ increases the weight by at least 2Δ. Since $j_{t-1} < k$, this argument holds for all $i \in \{1, \ldots, t-1\}$. □

In the following, we show that vertices which have potential less than a certain threshold cannot contribute significantly to a solution of the DkS problem, and can thus be removed from the graph.

4.2.1 Threshold-based Pruning

In this subsection, we give an algorithm which computes a so-called Δ-core which still contains a $(1 + \frac{1}{k-1})$-DkS approximate solution. At first, we give a definition.

Definition 4.4 (Weighted Δ-core)
Let $G \coloneqq (V, E, w)$ be a weighted graph and $\Delta \geq 0$. A set of vertices $C \subseteq V$ is a Δ-core of G if each vertex has potential (or weighted degree) no less than Δ in the subgraph induced by S:

$$p_{G(C)}(v) \geq \Delta \text{ for all } v \in C.$$

Algorithm 4.1 is an adaption of a well-known algorithm which determines an unweighted Δ-core, see [Łu91]. It computes a Δ-core C of a graph G by removing vertices from G in the following way: The set of residual vertices C is initialized as $C \leftarrow V$; T is the set of vertices which still have to be considered for removal and is initialized as $T \leftarrow V$. For each vertex $v \in T$, the initial potential $p(v)$ is the sum of weights of all adjacent edges (lines 2 – 4).

Input: Size of subgraph $k \in \mathbb{N}$, weighted graph $G \coloneqq (V, E, w)$, threshold Δ
Output: Δ-Core $C \subseteq V$

```
1  C ← V, T ← V
2  for v ∈ T do
3  |  p(v) ← Σ_{v'∈N_G(v)} w({v, v'})
4  end
5  while T ≠ ∅ and |c| > k do
6  |  choose vertex v ∈ T
7  |  T ← T \ {v}
8  |  if p(v) < Δ then
9  |  |  for v' ∈ N_{G(C)}(v) do
10 |  |  |  p(v') ← p(v') − w({v, v'})
11 |  |  |  T ← T ∪ {v'}
12 |  |  end
13 |  |  V' ← V' \ {v}
14 |  end
15 end
```

Algorithm 4.1: weighted-Δ-Core

Trivially, the potential is an upper bound on how much this vertex can contribute to the total edge weight of an induced k-vertex subgraph.

The main *while loop* (lines 5–14) is iterated until T contains no more vertices. In each of the iterations, we choose and remove an arbitrary vertex v from the set. If its potential is below the threshold (line 8), all of its neighbours v' are added to T, and their potentials are updated (lines 9–12). Since the neighbours of deleted vertices are added to T again, we obtain a chain-reaction of deletions: At first, only vertices with a potential less than our input threshold are deleted. Yet, with the deletion of a vertex, the potentials of some of the remaining vertices are reduced, and may drop below the threshold as well. A vertex incident to many vertices of low potential may be deleted at a late stage of the algorithm.

The vertices $v_1, \ldots, v_{k-1} \in \mathcal{S}$ are guaranteed to remain in G', since — by definition — each of them has potential no less than the threshold in the subgraph induced by $\mathcal{S}$ and therefore also in each graph containing $\mathcal{S}$.

The following lemma gives a lower bound for the absolute loss of edge weight with respect to a set of vertices that we may introduce by performing these vertex deletions.

Lemma 4.5
Let $G \coloneqq (V, E, w)$ be a weighted graph, $S \subseteq V$ a subset and $C \subseteq V$ the Δ-core obtained by Algorithm 4.1. Then we have

$$w(S \cap C) \geq w(S) - |S \setminus C| \cdot \Delta.$$

Proof. First, note that $t \coloneqq |S \setminus C|$ is the number of vertices in S that were deleted during the run of Algorithm 4.1. Let $\mathcal{D} \coloneqq (v_1, \ldots, v_t)$ be the ordered set of these vertices, in the order

in which they were deleted. We prove the claim by induction on $i = 1, \ldots, t$, by showing that

$$w\Big((S \cap C) \cup \{v_1, \ldots, v_i\}\Big) \leq w(S \cap C) + i \cdot \Delta.$$

For $i = 1$, we know that $w((S \cap C) \cup \{v_1\}) \leq w(S \cap C) + \Delta$, as $p_{G(C)}(v_1) < \Delta$. Let us assume that the claim holds for $i - 1$, and consider i. In the iteration in which vertex v_i is deleted during the course of Algorithm 4.1, its remaining potential $p(v_i)$ (line 8) satisfies $\Delta > p(v_i) \geq w(E_i)$, where

$$E_i \coloneqq \{\{v_i, v\} : v \in (S \cap C) \cup \{v_{i+1}, \ldots, v_t\}\}.$$

Informally, E_i consists of the edges that connect v_i to those vertices in $S \subseteq V$ which have not been deleted yet.

Thus, deleting vertex v_i after $v_1, \ldots, v_{i-1}$ have already been deleted imposes only an additional error of at most Δ, i. e.,

$$\begin{aligned} w\Big((S \cap C) \cup \{v_1, \ldots, v_i\}\Big) &\leq w\Big((S \cap C) \cup \{v_1, \ldots, v_{i-1}\}\Big) + \Delta \\ &\leq w(S \cap C) + (i-1) \cdot \Delta + \Delta \\ &= w(S \cap C) + i \cdot \Delta. \end{aligned}$$

□

Using Lemma 4.5, we are now ready to prove that the subgraph remaining after a run of Algorithm 4.1 still contains a $(1 + \frac{1}{k-1})$-approximate solution for the DkS problem.

Theorem 4.6
Let $G \coloneqq (V, E, w)$ be a weighted graph, $k \in \mathbb{N}$, and $\mathcal{S} \coloneqq (v_1, \ldots, v_k)$ a k-vertex ordered set in G with threshold Δ. Then Algorithm 4.1 computes a Δ-core $C \subseteq V$ which contains a $(1 + \frac{1}{k-1})$-approximate solution for the DkS problem.

Proof. Let $S_O \subseteq V$ be an optimal set of k vertices for the DkS problem on G and $S_C \subseteq C$ an optimal DkS vertex set for the subgraph $G(C) = (C, E(C))$ induced by C.

If $S_O \subseteq C$, we have $w(S_O) = w(S_C)$, and are done. Otherwise, there is a set $V_D \coloneqq S_O \setminus C \neq \emptyset$ of $t \in \mathbb{N}$ vertices of the optimal solution that were pruned. By Lemma 4.5, we lose at most total edge weight $t \cdot \Delta$ by removing these vertices from S_O, i. e., $w(S_O \cap C) \geq w(S_O) - t\Delta$. Let further $l \in \mathbb{N}$ and $j_1, \ldots, j_l \in \mathbb{N}$ such that $\mathcal{S} \setminus S_O = (v_{j_1}, \ldots, v_{j_l})$, where $j_s < j_{s'}$ for $s < s'$. Since $\mathcal{S} = (v_1, \ldots, v_k) \subseteq C$, we know that $V_D \subseteq S_O \setminus \mathcal{S}$ and thus $l = |\mathcal{S} \setminus S_O| = |S_O \setminus \mathcal{S}| \geq |V_D| = t$.

Using $S_O \cap \mathcal{S} \subseteq S_O \cap C$, this implies

$$\{v_1, \ldots, v_{j_t}\} \subseteq (S_O \cap C) \cup \{v_{j_1}, \ldots, v_{j_t}\} \text{ and } \{v_1, \ldots, v_{j_t}\} \subseteq \mathcal{S}.$$

Applying Lemma 4.3 to the ordered set $(v_1, \ldots, v_{j_t})$ with threshold

$$\Delta((v_1, \ldots, v_{j_t})) \geq \Delta(\mathcal{S}) = \Delta$$

yields

$$w((S_O \cap C) \cup \{v_{j_1}, \ldots, v_{j_t}\}) \geq w(S_O \cap C) + (t-1)\Delta \geq w(S_O) - \Delta.$$

The worst possible error is the additive absolute error of Δ. Note that

$$(S_O \cap C) \cup \{v_{j_1}, \ldots, v_{j_t}\} \subseteq C$$

and thus $w(S_C) \geq w(S_O) - \Delta$. As $w(S_C) \geq w(\{v_1, \ldots, v_k\}) \geq (k-1) \cdot \Delta$, the relative error is bounded from above by $1 + \frac{1}{k-1}$:

$$\frac{w(S_O)}{w(S_C)} \leq \frac{w(S_C) + \Delta}{w(S_C)} \leq 1 + \frac{\Delta}{(k-1)\Delta} = 1 + \frac{1}{k-1},$$

which proves the claim. □

Finally, let us briefly turn to the running time of Algorithm 4.1.

Theorem 4.7
Let $\Delta \geq 0$ and $G := (V, E)$ be a graph with $n := |V|$ and $m := |E|$. There is an implementation of Algorithm 4.1 with running time $\mathcal{O}(m + n)$.

Proof. The initialization of the potentials in lines 2–4 is possible in time $\mathcal{O}(m + n)$, as each edge only contributes to the potentials of the two incident vertices. Initially, T contains $n - k$ vertices, and at most m edges can be removed by the algorithm, as a result of losing one of their incident vertices. Due to this, the *while loop* (lines 5–14) is iterated at most $n - k + m$ times.

The *for loop* in lines 9–12 is performed at most m times during a run of the algorithm, initiated by the deletion of an edge. It consists of a single subtraction, and two set operations. This proves the claim. □

For the sake of a simple notation, we used a simplified definition of the potential in our proofs and in Algorithm 4.1. In fact, a better way to define the potential of vertices is to only count the $k - 1$ largest incident edges. Note that we then have to invest additional effort to identify whether the removed edge was one of the $k-1$ largest ones or not. This information also has to be updated. In practice, this extra effort is worthwhile, as it allows for the deletion of more vertices, while the approximation ratio is maintained. In many cases, it is a much better (i. e., lower) estimate for how much a specific vertex can contribute to the total edge weight of an induced k-vertex subgraph in a best-case scenario.

4.2.2 Approximate Solution

By combining threshold-based pruning with an exact algorithm, we obtain a (non-polynomial time) algorithm for the DkS problem with approximation error of at most $(1 + \frac{1}{k-1})$ in the form of Algorithm 4.2. We determine an ordered set of vertices and a corresponding threshold and use this threshold in an application of Algorithm 4.1 to reduce the size of the graph G. On the reduced graph, we use a traditional branch-and-cut algorithm or solve a quadratic model of the DkS problem.

Input: Size of subgraph $k \in \mathbb{N}$, weighted graph $G := (V, E, w)$
Output: k-vertex set S_C which is a $(1 + \frac{1}{k-1})$-DkS approximation for G

1 $\mathcal{S} \leftarrow$ `maxThresholdSet`(G)
2 $C \leftarrow$ `weighted-Δ-Core`$(G, \Delta(\mathcal{S}))$
3 $S_C \leftarrow$ `dense`k`Subgraph`$(G(C), k)$

Algorithm 4.2: approximateDkS

Theorem 4.8
Let $G \coloneqq (V, E, w)$ be a weighted graph and $k \in \mathbb{N}$. Then Algorithm 4.2 computes a $(1 + \frac{1}{k-1})$-approximation for the DkS problem.

Proof. In line 3, the algorithm solves the DkS problem exactly for the graph $G(C)$ induced by the Δ-core C. This implies that it suffices to show that $G(C)$ still contains a $(1 + \frac{1}{k-1})$-approximate solution. Thus the claim follows by Theorem 4.6. □

The bound given in Theorem 4.8 is also best possible. This is demonstrated by the worst case example given in Figure 4.3. For $k \coloneqq 3$, $\mathcal{S} \coloneqq (v_1, v_2, v_3)$ is an ordered set with (optimal) threshold $\Delta = 2 + \varepsilon$. The Δ-core obtained by Algorithm 4.1 is $C \coloneqq \{v_1, \ldots, v_4\}$. An optimal solution to Problem DkS for $k = 3$ on C is $S_C \coloneqq \{v_1, v_2, v_3\}$ with weight $4 + 2\varepsilon$. The optimal solution to the problem on G is $S_O \coloneqq \{v_3, v_4, v_5\}$ with objective value 6.

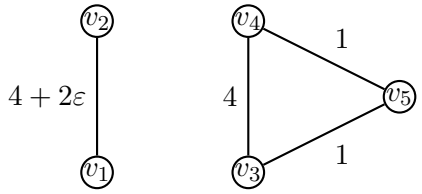


Figure 4.3: Worst case example for the approximation ratio Algorithm 4.1. For arbitrarily small $\varepsilon > 0$ and $k = 3$, the algorithm removes vertex v_5, and a densest 3-subgraph of the resulting graph $G(C)$ has weight $4 + 2\varepsilon$.

Generally, the running time of the algorithm is dominated by the computation of an exact solution to the problem for the reduced graph in line 3, and thus depends greatly on the size of the threshold computed in line 1. The next section is dedicated to the computation of a provably optimal threshold in polynomial time.

4.3 An Optimal Threshold

In this section, we devise a greedy-like algorithm for the construction of an ordered vertex set with optimal threshold. Consider Algorithm 4.3, which describes the approach in pseudo-code.

The ordered set $\mathcal{S}$ is initialized with the largest edge in the graph (line 1). For all vertices which are not covered by $\mathcal{S}$, we compute their potential $p(i)$ with respect to $\mathcal{S}$ (lines 2–4). This value represents the difference $w(\mathcal{S} \cup \{v_i\}) - w(\mathcal{S})$. Then, in lines 5–7, we compute the $\mathcal{S}$-potentials $p(i, j)$, which represent the differences $w(\mathcal{S} \cup \{v_i, v_j\}) - w(\mathcal{S})$. This value is the sum of the potentials $p(i), p(j)$ and of $w(\{v_i, v_j\})$.

In the main *while loop* (lines 8–27), we iteratively add vertices to $\mathcal{S}$. To do so, we find a vertex v_i with highest $\mathcal{S}$-potential $p(i)$ and the pair of vertices v_j, v_l with maximal $p(j, l)$. If v_i contributes more potential than $\frac{1}{2}p(j, l)$, we append v_i to the end of the ordered set and update the potentials of the other vertices in $V \setminus \mathcal{S}$ according to the edges incident with v_i (lines 12–17). Otherwise, i. e., if appending v_j and v_l yields at least twice the potential of $p(i)$, we add these two vertices to the end of $\mathcal{S}$. When appending v_j, v_l to $\mathcal{S}$, it does not matter whether v_j or v_l is added first. Again, the potentials of the other vertices in $V \setminus \mathcal{S}$ are updated (lines 18–23). In both cases, we also update the joint potentials of all pairs of vertices not in $\mathcal{S}$ (lines 24–26).

Note that by comparing the addition of a best single vertex with that of a best pair of vertices, the operations in the main loop are tailored towards providing a greedy currently-best value δ^i in the constructed ordered set.

After leaving the *while loop*, $\mathcal{S}$ contains either k or $k-1$ vertices. If it contains k vertices, we are done. Otherwise, we add a final vertex, best with respect to $p(i)$. The ordered set $\mathcal{S}$ and threshold Δ are returned.

Input: Size of subgraph $k \in \mathbb{N}$, weighted graph $G \coloneqq (V, E, w)$ with $w\colon E \longrightarrow \mathbb{N}$
Output: Ordered set $\mathcal{S} \subseteq V$ with maximal threshold Δ

Find maximum-weight edge $\{v_i, v_j\} \in E$, $\mathcal{S} \leftarrow (v_i, v_j)$, $\Delta \leftarrow \frac{1}{2}w(e_1)$
for $v_i \in V \setminus \mathcal{S}$ **do**
  $p(i) \leftarrow \sum_{v \in \mathcal{S}} w(\{v_i, v\})$
end
for $v_i, v_j \in V \setminus \mathcal{S}$ *with* $i < j$ **do**
  $p(i,j) \leftarrow p(i) + p(j) + w(\{v_i, v_j\}$
end
while $|\mathcal{S}| \leq k-2$ **do**
  Find $v_i \in V \setminus \mathcal{S}$ with maximal $p(i)$
  Find $v_j, v_l \in V \setminus \mathcal{S}$ with $j < l$, with maximal $p(j,l)$
  $\Delta_1 \leftarrow p(i)$, $\Delta_2 \leftarrow \frac{1}{2}p(j,l)$
  if $\Delta_1 > \Delta_2$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{v_i\}$, $\Delta \leftarrow \min\{\Delta, \Delta_1\}$
    for $v'_i \in V \setminus \mathcal{S}$ **do**
      $p(i') \leftarrow p(i') + w(\{v_i, v_{i'}\})$
    end
  end
  else
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{v_j, v_l\}$, $\Delta \leftarrow \min\{\Delta, \Delta_2\}$
    for $v'_i \in V \setminus \mathcal{S}$ **do**
      $p(i') \leftarrow p(i') + w(\{v_j, v_{i'}\}) + w(\{v_l, v_{i'}\})$
    end
  end
  for $v_{i'}, v_{j'} \in V \setminus \mathcal{S}$ *with* $i' < j'$ **do**
    $p(i', j') \leftarrow p(i') + p(j') + w(\{v_{i'}, v_{j'}\})$
  end
end
if $|\mathcal{S}| = k-1$ **then**
  Find $v_i \in V \setminus \mathcal{S}$ with maximal $p(i)$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{v_i\}$
end

Algorithm 4.3: maxThresholdSet

We prove that Algorithm 4.3 computes an optimal threshold for a given graph.

Theorem 4.9
Let $k \in \mathbb{N}$ and $G := (V, E, w)$ be a weighted graph. Algorithm 4.3 computes an ordered set $\mathcal{S}_g$ of k vertices with maximal threshold among all ordered sets of k vertices of the graph.

Proof. Let $\mathcal{S}_o := (w_1, \ldots, w_k) \subseteq V$ be an ordered set with maximal threshold $\Delta(\mathcal{S}_o)$ and let $\mathcal{S}_g := (v_1, \ldots, v_k)$ be the ordered set computed by Algorithm 4.3 with threshold $\Delta(\mathcal{S}_g)$. Suppose $\Delta(\mathcal{S}_g) < \Delta(\mathcal{S}_o)$ and let $i \in \{1, \ldots, k-1\}$ be the minimal index such that $\delta_g^i < \Delta(\mathcal{S}_o)$ and $\mathcal{S} := (w_{j_1}, \ldots, w_{j_t})$ the ordered set obtained from $\mathcal{S}_o$ by removing all occurrences of v_j for all $j \in \{1, \ldots, i-1\}$.

Since $i - 1 \leq k - 2$, $\mathcal{S}$ contains at least two vertices and $(w_1, \ldots, w_{j_2})$ is an ordered subset of $\mathcal{S}_2 := (v_1, \ldots, v_{i-1}, w_{j_1}, w_{j_2})$. Therefore $\delta_2^i \geq \delta_o^{j_1} \geq \Delta(\mathcal{S}_o)$. This is a contradiction to the greedily-best choice of vertices in the *while loop* of the algorithm. □

Next, let us consider the running time of an implementation of the algorithm along the lines of the given pseudo-code.

Theorem 4.10
Let $G := (V, E, w)$ be a weighted graph with $n := |V|$, $m := |E|$ and $k \in \mathbb{N}$. There is an implementation of Algorithm 4.3 with running time $\mathcal{O}(kn^2)$.

Proof. Finding the largest edge in the graph and computing the potentials of the vertices can be realized in $\mathcal{O}(m + n)$, as each edge has to be considered at most twice. Initialization of the potential of the edges is in $\mathcal{O}(n^2)$, as each pair of vertices and each edge have to be considered at most once.

Further, each iteration of the main *while loop* can be achieved in $\mathcal{O}(n^2)$: finding the maximal $p(i)$ is in $\mathcal{O}(n)$, finding the maximal $p(i, j)$ is in $\mathcal{O}(n^2)$, and the potential-updates are possible in $\mathcal{O}(n^2)$. In each iteration, at least one vertex is appended to the ordered set. Thus, the *while loop* is iterated at most $k - 1$ times resulting in time $\mathcal{O}((k-1)n^2)$ for the operations of the *while loop*. Appending a final vertex to the set is possible in $\mathcal{O}(n)$.

In total, the running time is dominated by the initialization ($\mathcal{O}(n^2)$) and the *while loop* ($\mathcal{O}((k-1)n^2)$), yielding a total running time of $\mathcal{O}(kn^2)$. □

By Theorems 4.9 and 4.10, Algorithm 4.3 is an efficient way of computing a provably optimal threshold which is necessary for Algorithm 4.2. For graphs with properties like those occurring in our practical application in forestry, this approach works very well.

4.4 Empiric Results

We applied Algorithm 4.2 on a total of 62 forestry regions in Northern Bavaria. The graphs $G := (V, E)$ — constructed as described in Section 1.2 — range from $n = 64$ to 684 vertices. An interesting property of these graphs is that their number of edges is typically bounded from above by $10n$. This small number comes from the special way in which they are constructed:

Recall that there is an edge in between two vertices in our graph if the two owners corresponding to these vertices have lots that share a common boundary. The lot structure of forestry regions allows us to model these neighbour-relations as a planar graph from which the graph in question can be derived by identification of vertices. Each lot is represented by its centre of gravity (or any arbitrary point in its interior), and there is an edge in between two such points if the corresponding lots share a common boundary. Let s be the number of lots

in the region. This planar graph then contains at most $3s - 6$ edges, which is also an upper bound on the number of edges $|E|$.

The number s is indirectly connected to $n := |V|$, as in most regions about 30–40% of the lots are owned by owners that only have this single lot. On average for all regions, each owner has about 5 lots, i. e., $n \approx \frac{1}{5}s$. Note further that many neighbourhood relations are in between the same owners, so that the lengths of these boundaries contribute to the weight of a single edge, and do not induce multiple edges. In practice, one arrives at the typical number of edges mentioned above. Recall that, by this construction, this graph does not satisfy the triangle inequality.

We performed Algorithm 4.2 for these graphs on a standard laptop[3]. Figure 4.4 depicts the result of such a DkS approximation for $k = 10$, by only showing the selected k owners in colour. The result is as expected: Some of the selected owners have many and very large lots, e. g., those depicted in brown and purple. These are complemented by owners that have a long common boundary with these owners' lots, as well as lots with bad shape. Good examples for this are the yellow and green owner (in the centre of the figure and to the upper right, respectively).

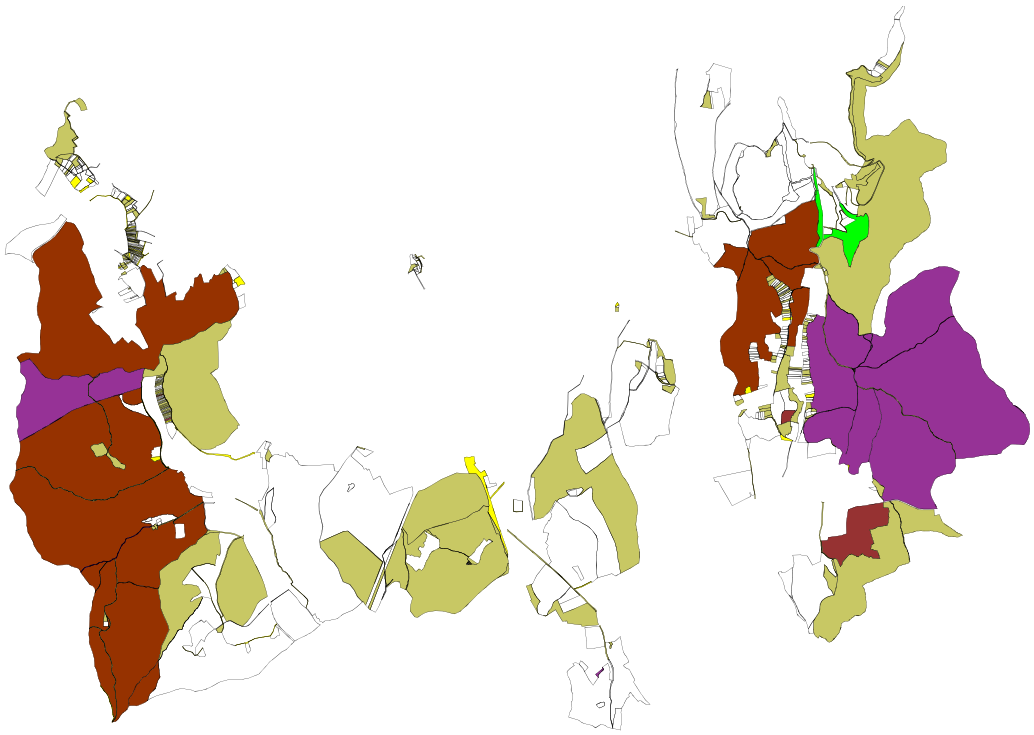


Figure 4.4: The forestry region of Figure 4.1. Only the $k := 10$ owners selected by a run of Algorithm 4.2 are coloured.

In these computations, the running time of our threshold-based pruning step never exceeded 1 second. In general, the total running time of Algorithm 4.2 depends almost solely on the final step, and thus on the number of vertices remaining after pruning. In other words, it depends greatly on the size of the computed threshold and the number of vertices whose potential falls

[3]Intel Dual Core i7 CPU at 2.0 GHz and 4 GB of RAM; algorithms implemented in Java 1.6

below this threshold. All of these vertices of low potential are pruned during a first iteration of Algorithm 4.1. Many of the vertices with a rather large original potential can be removed during later iterations of the algorithm.

The key reason for the success of this approach for the graphs in this real-world application is depicted in Figures 4.5 and 4.6. The black graph depicts the cumulative distribution of edge weights and potentials in a representative selection of regions, i. e., the percentage of edge weights and potentials less than or equal to edge weight w and potential p, respectively. In Figure 4.6, the corresponding threshold is illustrated by a black vertical line. A cumulative exponential distribution fitted to the data via a maximum likelihood estimator is shown in red and reveals a tight correspondence.

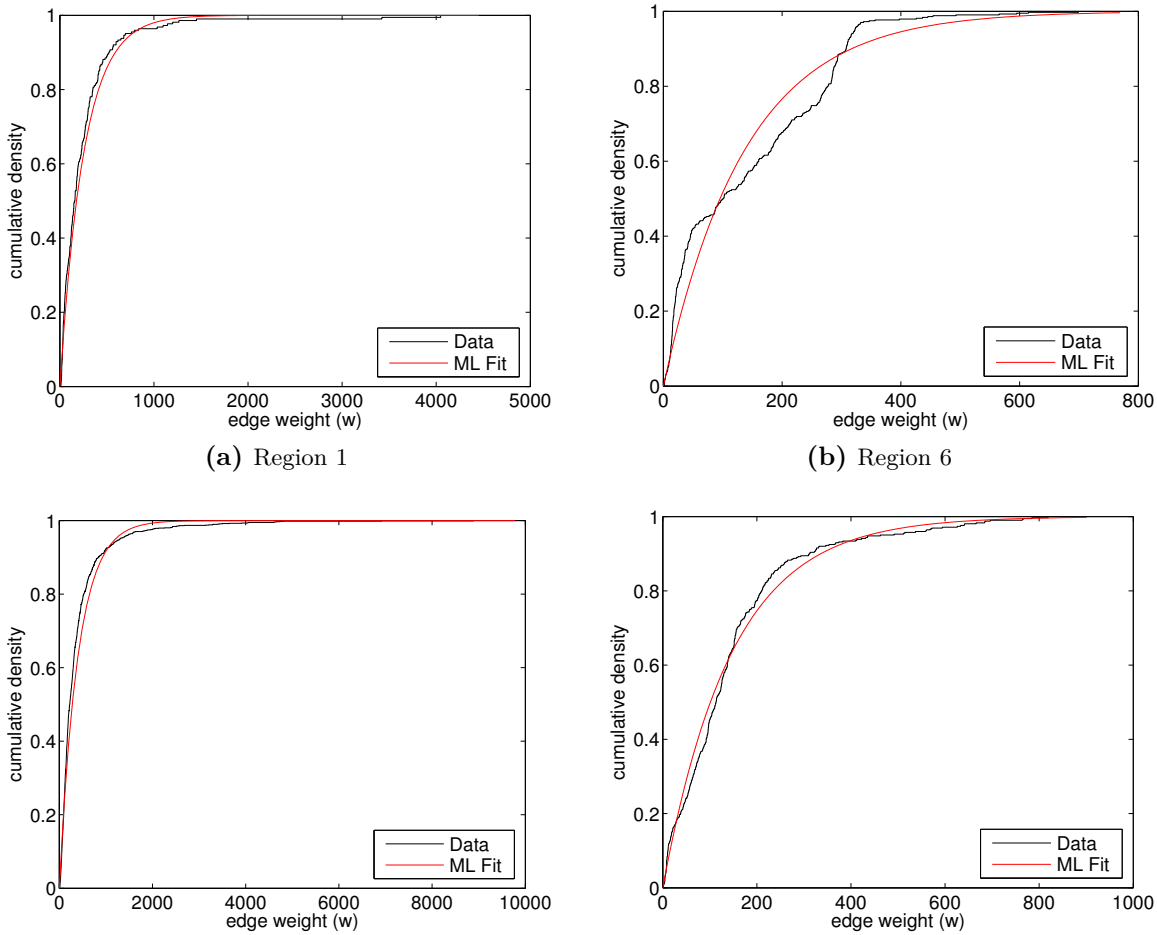
region	n	m	n_P	n_P/n	m_P	m_P/m
1	100	283	22	22,0%	98	34.6%
2	236	739	12	5.1%	44	6.0%
3	540	3258	40	7.4%	253	7.8%
4	469	1689	10	2.1%	27	1.6%
5	597	6107	13	2.2%	36	0.6%
6	155	484	128	82.6%	392	81%
7	163	378	48	29.4%	77	20.4%
8	347	1904	24	6.9%	104	5.5%
9	145	555	31	21.4%	161	29.0%
10	477	1402	10	2.1%	30	2.1%
11	212	834	13	6.1%	49	5.9%
12	315	2445	10	3.2%	38	1.6%
13	454	2467	11	2.4%	32	1.3%
14	395	1650	23	5.8%	199	12.1%
15	684	2678	46	6.7%	443	16.5%
16	669	5755	140	20.9%	856	14.9%

Table 4.1: Results of Algorithm 4.2 for a representative choice of forestry regions. The values n and m denote the number of vertices and edges of the original graph. The number of residual vertices and edges in the Δ-core computed by Algorithm 4.2 is given by n_P and m_P, respectively.

This "exponential character" of the distribution of edge weights is a key feature for the success of our pruning approach. Even low thresholds with respect to the maximal potential already cut off a significant portion of the graphs. Recall also that due to the iterative nature of our pruning approach, many vertices whose potential in the original graph is much higher than the threshold can be removed at a later stage of the algorithm, if their potential has decreased enough. Therefore, in practice, we remove a lot more vertices than indicated by the black line in Figure 4.6.

For 55 of our 62 instances, using $k := 10$, the graph-size reduction steps performed in Algorithm 4.2 yield a graph $G' := (V', E')$ for which our final exact computation terminated in at most 1 minute. For 13 of these 55 successful instances, our pruning algorithms already reduced the final graph to vertex-size exactly k, so that the final step was not entered at all. In most other cases, we obtained a reduction of the number of vertices in between 90% and 95%.

When exceeding a running time of 1 minute (for the 7 remaining instances), we returned



(a) Region 1 **(b)** Region 6

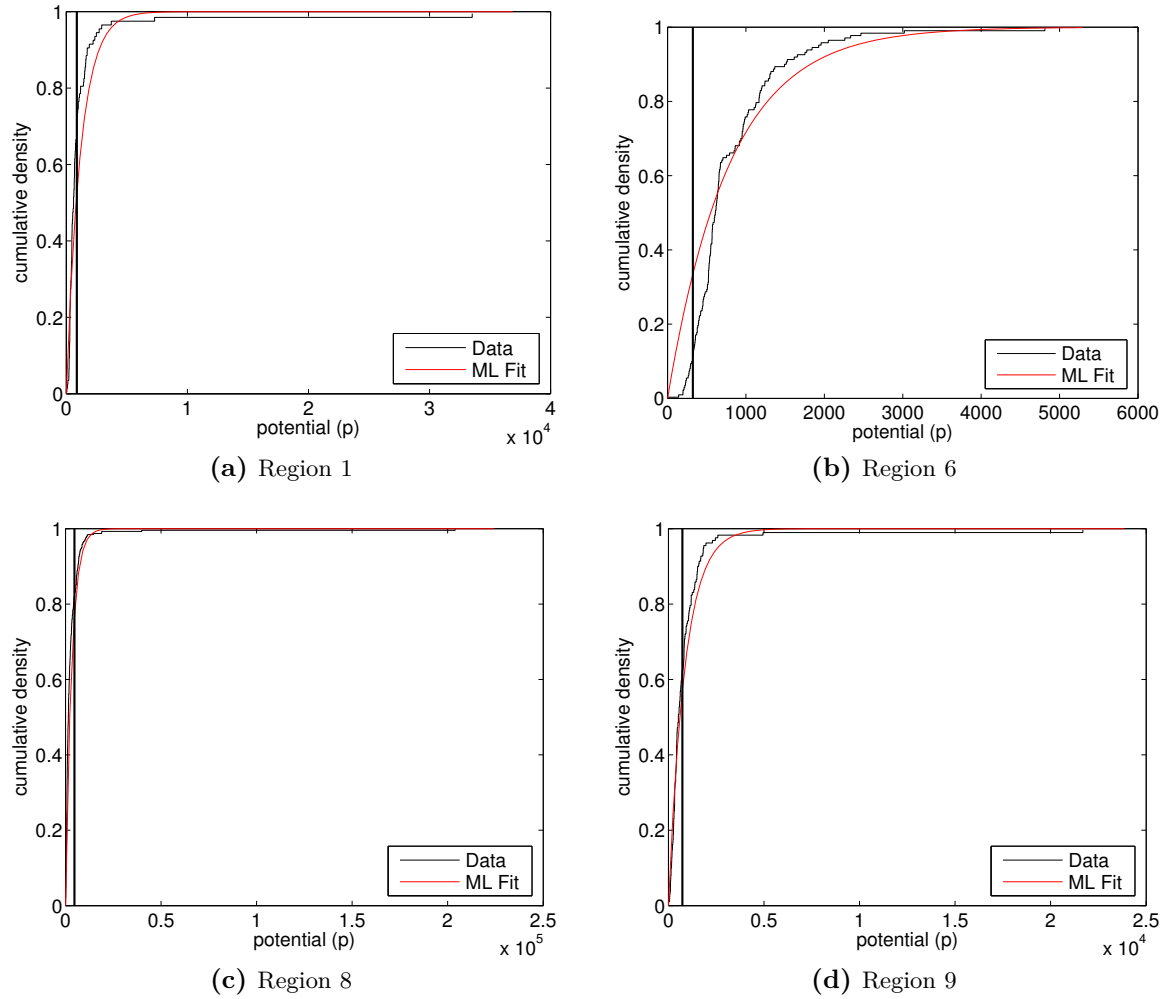
(c) Region 8 **(d)** Region 9

Figure 4.5: Cumulative distribution of edge weights for some representative forestry regions.

the greedy solution computed by Algorithm 4.3 for practical purposes. Even in these cases the a posteriori approximation error was less than 2.5.

Table 4.1 sums up the reduction of the number of vertices of the graphs for 16 anonymized representative regions. The original number of owners is denoted by n, while n_P denotes the number of owners after the pruning step, m the number of edges in the original graph, and m_P the number of edges in the induced subgraph after pruning. Note that for 14 of these 16 regions, we ended up with at most 48 vertices, so that the value of n_P was small enough for an exact solution of the problem for the reduced graph.

In conclusion, the empiric results exhibit that the special properties of the graphs derived from our application in forestry allow our threshold-based approximation algorithm to solve the DkS problem with a very good approximation ratio for almost all of our instances, despite their large number of vertices. While one cannot expect similar numbers for general graphs, our pruning steps are computationally cheap, and may prove helpful in many applications. In general, the total running time is dominated by the solution of the reduced problem, for which a sophisticated implementation along the lines of state of the art literature is essential.



(a) Region 1 **(b)** Region 6

(c) Region 8 **(d)** Region 9

Figure 4.6: Cumulative distribution of potentials some for representative forestry regions.

Unfortunately, the graphs arising from the reformulation of Problem p-BGHD do not possess similar structure. Usually, each point pair has low distortion to at least some other point pairs, which overall results in a graph with roughly equally distributed weights. This is in contrast to the assumptions which the algorithms presented above are based on and which are fulfilled in the application in forestry. It is therefore not surprising that the approach above does not yield satisfying results in this setting. The homogeneous distribution of weights causes a low threshold and in turn only few vertices can be pruned. An exact solution on the reduced graph is also complicated by the huge size of the original instances in question. By construction, the graph representing the problem for metric spaces (X, d_X) and (Y, d_Y) has cardinality $|X||Y|$. This is far beyond the capabilities of the exact methods for Problem DkS for the metric spaces we consider in Chapter 3, even after the reduction achieved with our pruning approach.

A generalization of the approach presented here could be realized by considering the $l \in \mathbb{N}$ next vertices in each step of the greedy construction of a sequence of maximal threshold, instead of $l := 2$. The definition of the threshold could be adapted as well, allowing for a higher threshold and thus smaller reduced graphs.

Chapter 5

Mesh Segmentation

Mesh segmentation can be formulated as a class of optimization problems with constraints and objective functions adapted for various applications. In most cases, the constraints and objective functions are formulated in terms of features of the mesh, which can be classified in several groups. The survey [Sha08] gives an overview of commonly used methods and identifies three major types of constraints:

Cardinality constraints confine the number of faces (or vertices) in a partition or the number of segments in the partition.

Geometric constraints control geometric properties such as area and diameter but also convexity and curvature of the segments.

Topological constraints require topological properties such as connectivity or equivalence to certain objects, e. g., a disc.

Similarly, there are many properties of a mesh which can be of interest for a "good" segmentation and therefore influence the objective function. Examples for such properties are planarity, curvature, geodesic distances, symmetry, convexity, and many others.

An intuitive approach to segmentation, if the number of segments is known a priori, is using methods based on clustering algorithms, specifically the so-called Lloyd's or k-means algorithm [Llo82]. Shlafman, Tal, and Katz [STK02] use centroids of the faces and their surface normals to define distances between faces of an orientable mesh, while Cohen-Steiner, Alliez, and Desbrun [CSAD04] and Kobbelt and Wu [KW05] consider the distance of segments to given reference shapes — so-called shape proxies — to perform k-means clustering on the resulting space.

A common way to perform segmentation of a mesh if the number of segments is unknown is by so-called region growing methods, which are also commonly used in the related task of mesh simplification, see e. g., [KT96; LDB05; Cha+95]. After determining some faces which constitute initial regions in a first step, these regions are increased by adjoining adjacent faces. The choice of the faces which are adjoined is usually based on a greedy evaluation of the objective function, i. e., on (some of) the properties mentioned above.

A similar approach is taken by hierarchical clustering as introduced in [GWH01]. In its basic form, the algorithm tries to identify large plane areas of the mesh. This is achieved by a greedy heuristic operating on the dual graph of the mesh. In each step, the parts of the mesh which admit the best approximation by a plane are joined. An advantage of this method is that one run of the algorithm yields segmentations with a decreasing number of segments (but increasing approximation error). Therefore, a suitable number of segments can be chosen a posteriori such that constraints on the approximation error are respected. The

algorithm has been extended to consider approximation of segments not only by planes, but also by several other primitives, cf. [AFS06]. For each join of the mesh parts, one computes (or estimates) the error of approximating the resulting part by the given primitives. Note that one step of the algorithm requires that all possible join operations are evaluated by fitting each given primitive to the resulting part. To be usable in practice, this requires very fast methods for shape matching, i. e., for the fitting of these primitives. Consequently, only select primitives which admit efficient algorithms for the shape matching procedure have been considered by the authors. Investigations with respect to suitable primitives and respective shape matching methods have also been considered in [Woh10]. Due to the very limited choice of such primitives, this approach is not viable for the goal of the project.

5.1 Minimal Covering of a Mesh

In the preceding chapters, we developed the tools for recognizing parts of the component which can be manufactured by elements in our database. In practice, however, this will not result in the desired segmentation of the component. If the database is of reasonable size, we can expect that the surface can be covered by several sets of standard elements. Thus, the next step towards a segmentation is selecting a number of parts which are actually used to construct the component.

To achieve this, we compute a minimal covering of the surface, i. e., a subset of the standard elements that minimizes unnecessary overlap while still covering the whole component surface. We model this problem as a variant of the weighted dominating set problem.

Recall that both the component surface and the standard elements are represented as meshes. We construct a graph by assigning a so-called *terminal* node to each standard element and a node to each point of the mesh. Each terminal is connected to all nodes that are covered by the corresponding standard element by a so-called *tn-edge*. In addition, we connect any two terminals whose corresponding standard elements overlap by an edge we refer to as *tt-edge*. Each tt-edge is assigned a weight proportional to the overlapping area. Our goal is then to find a subset of the terminals that dominates the nodes, i. e., where each node is adjacent to at least one selected terminal, while minimizing the total edge weight between selected terminals. This corresponds to finding a dominating set G with minimal induced edge weight.

Problem 5.1 (Minimal dominating set (MDS))
Input: Graph $G \coloneqq (V, E)$, set of terminals $T \subseteq V$, weight function $w\colon E(T) \longrightarrow \mathbb{N}$, $L \in \mathbb{N}$.
Question: Is there a dominating set $D \subseteq T$ of G with induced weight $w(E(D)) \leq L$?

At first, we investigate the computational complexity of this problem.

Theorem 5.2
Problem MDS cannot be approximated in polynomial time within a factor of $\mathcal{O}\left(2^{\mathrm{poly}(\langle I\rangle)}\right)$, *where* $\langle I\rangle$ *denotes the size of an instance* I *of Problem MDS, unless* $\mathcal{P} = \mathcal{NP}$.

Proof. We use an argument similar to the one employed in [Hal93] to establish the hardness of approximating the minimal size of an independent set which cannot be extended. We show that an approximation algorithm for Problem MDS can be used to decide the $\mathcal{NP}$-complete satisfiability problem SAT, cf. [GJ79, p. 259].

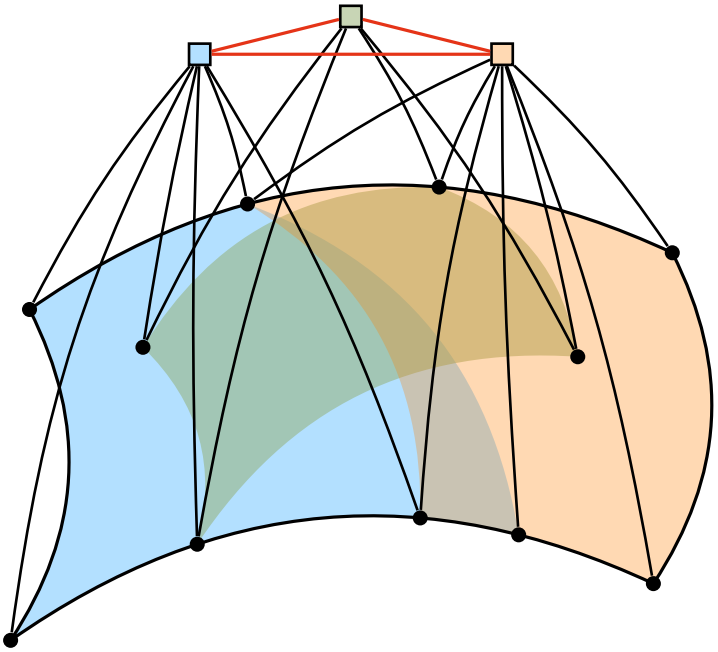


Figure 5.1: Minimal covering: construction of the graph for the dominating stable set problem. Terminals are represented by square nodes with colour corresponding to the standard element, mesh nodes as black circles. The tt-edges are drawn in red, while tn-edges are depicted in black.

Problem 5.3 (Boolean satisfiability (SAT))
Input: Variables $x_1, \ldots, x_n$, clauses $C_1, \ldots, C_m$
Question: Is there a truth assignment $\{x_1, \ldots, x_n\} \longrightarrow \{true, false\}$ such that

$$C_1 \wedge \ldots \wedge C_m = true?$$

Given a SAT formula F with n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$, we construct a graph $G \coloneqq (V, E)$ with $2n + m$ nodes $V \coloneqq C \dot\cup T$, where the m nodes $C \coloneqq \{C_1, \ldots, C_m\}$ represent the clauses and $2n$ terminals $T \coloneqq \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$ represent the literals. We construct the graph such that the subgraph induced by the terminals is complete, and each clause is connected to the nodes representing the literals it consists of, i. e.,

$$E \coloneqq \{\{\hat{x}, \hat{x}'\} \mid \hat{x}, \hat{x}' \in T\} \cup \{\{\hat{x}, C_k\} \mid \hat{x} \in T,\ C_k \in C \text{ and } C_k \text{ contains } \hat{x}\}.$$

Further, consider the weight function $w \colon E(T) \longrightarrow \mathbb{N}$ with

$$\begin{aligned} w(\{x_i, \bar{x}_i\}) &\coloneqq M, && \text{for all } i \in \{1, \ldots, n\} \\ w(\{x_i, x_j\}) &\coloneqq 1 =: w(\{x_i, \bar{x}_j\}), && \text{for all } i, j \in \{1, \ldots, n\}, i \neq j. \end{aligned}$$

for a sufficiently large $M \in \mathbb{N}$.

Each truth assignment $f \colon \{x_1, \ldots, x_n\} \longrightarrow \{true, false\}$ of the variables corresponds to a subset $D_f \coloneqq \{x_i \in T \mid f(x_i) = true\} \cup \{\bar{x}_i \in T \mid f(x_i) = false\}$ of terminals and vice versa.

If f is a satisfying assignment for F, then D_f is a dominating set of G, since each clause contains at least one literal with value *true*. Since f is an assignment for F, D_f cannot contain the positive and negative literal of a variable, and therefore the weight of the graph induced by D_f is bounded by $w(E(D_f)) \leq \frac{2n(2n-1)}{2} = n(2n-1)$.

A dominating set $D \subseteq S$ of G of weight $w(E(D)) < M$ cannot contain both the positive and negative literal of a variable and is therefore a satisfying assignment for F.

Since M was arbitrary and $\langle M \rangle = \lceil \ln(M) \rceil$, this proves the claim. □

For the solution of practical instances emerging in the engineering application, we developed and implemented an integer linear programming approach. The set of mesh nodes and terminals are denoted by V_M and T, respectively. Since the number of standard elements which can be fitted to the part, i. e., the number of terminals, is small in practice, the algorithm performed efficiently in our tests. In addition, the formulation can also be adapted to handle slight modifications, e. g., allow for small uncovered surface parts.

$$
\begin{aligned}
\min \quad & \sum_{\{i,j\} \in E(T)} c_{ij} x_{ij} \\
\text{s.t.} \quad & x_{ij} \geq s_i + s_j - 1 && \forall \{i,j\} \in E(T) \\
& \sum_{s \in N_G(v)} s \geq 1 && \forall v \in V_M \\
& s \in \{0,1\}^{|T|} \\
& x \in \{0,1\}^{|T| \times |T|}
\end{aligned}
$$

The variables x_{ij} represent edges between the terminals. The corresponding edge weights c_{ij} are given by the amount of overlap of the two transformed standard elements. The variables s_i are decision variables for the terminals and represent which standard elements will be used to cover the surface. The constraints ensure that an edge between two terminals must be chosen if both terminals are chosen and that each vertex of the mesh must be adjacent to at least one terminal.

5.2 Subdivision in Submeshes

Finally, each of the remaining overlapping surface areas has to be split between the covering standard elements. This subdivision problem can be modelled as a so-called multi-terminal cut problem. To obtain a mesh segmentation, we need to assign each face to exactly one part. We will therefore consider the dual graph $G_D := (V_D, E_D)$ of the mesh, where V_D is the set of faces of the mesh, and two faces are connected by an edge $e \in E_D$ if and only if they intersect in a common mesh edge. We will refer to the vertices of this graph as nodes and the edges as *nn-edges*.

For the subdivision problem, we only have to consider faces of the mesh that are covered by more than one of the selected standard elements. For each standard element involved, we create a terminal node and each face node is connected to all terminals corresponding to standard elements that cover the face. These edges are again referred to *tn-edges* in the following. To represent the difficulty or cost of assigning two adjacent faces to different standard elements, we

assign a weight to the corresponding nn-edge. The cost of separating adjacent faces is, on the one hand, influenced by the curvature and material properties of the corresponding standard elements, since these affect the difficulty of producing such a cropped standard element. For example cropping along the directions of principal curvature is preferred. On the other hand, the quality of the fit of the cropped standard elements determines the difficulty of assembly of these parts and is thus also relevant for the computation of the edge weight.

The weight of a tn-edge is a measure for the quality of the alignment of the face corresponding to the node and the standard element represented by the terminal. If the error of the alignment of the corresponding parts is high, the weight on the corresponding edge is low and vice versa. Therefore, faces which are well aligned with the target geometry are preferred.

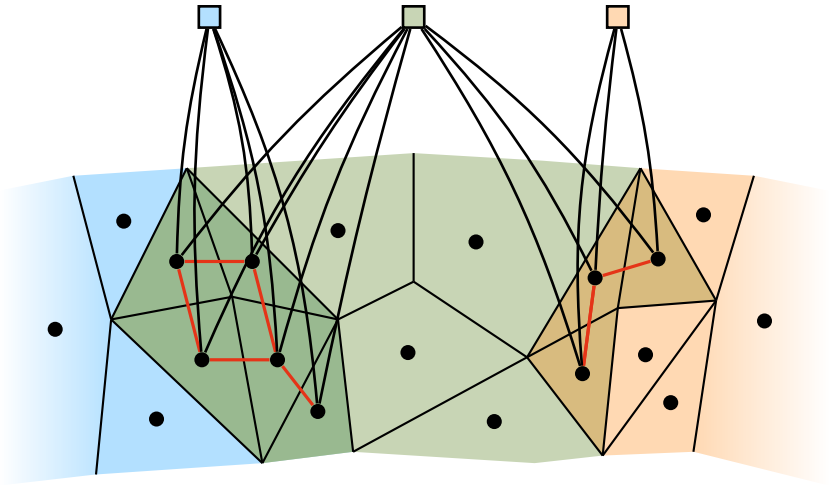


Figure 5.2: Subdivision: construction of the graph for the multi-terminal cut problem. The nn-edges are depicted in red, while the tn-edges are depicted in black. Terminals are represented by square nodes with colour corresponding to the standard element and nodes representing a face are drawn as black circles.

The problem is then to remove as few edges as possible with respect to their total weight such that the terminals become disconnected. The solution yields a unique mapping of each mesh face to one of the standard elements, hence all overlap is removed.

By the construction of the graph above, this is exactly a multi-terminal cut problem.

Problem 5.4 (Multi-terminal cut (MTC))
Input: Weighted graph $G := (V, E, w)$, set of terminals $T \subseteq V$, $L \in \mathbb{N}$.
Question: Is there a cut $C \subseteq E$, i. e., a subset of edges such that no two terminals are in a common component of $(V, E \setminus C)$, with weight $w(C) \leq L$?

Dahlhaus et al. [Dah+94] establish that the problem is $\mathcal{NP}$-hard for general graphs and $|T| \geq 3$ and also give a $\left(2 - \frac{2}{|T|}\right)$-approximation algorithm. Furthermore, Xiao [Xia08] shows that the problem is fixed parameter tractable with respect to the parameter $|T|$.

In practice, the overlapping areas of elements are rather limited. As already mentioned, the number of standard elements is usually small as well, such that the instances of this problem are not too large in our application. Therefore, the methods mentioned above suffice to produce satisfactory solutions in reasonable running time.

5.3 Conclusion

The methods described in this thesis are a first step towards the fully automated production of almost arbitrary target geometries from sheet metal parts. In conjunction with the results obtained in the underlying engineering project (see Chapter 1), we show that the segmentation of a target geometry in suitable standard elements is a viable approach to this challenging task. Due to the complexity of the segmentation problem, we employ a three stage approach consisting of shape matching, minimal covering and subdivision of the target geometry, cf. Section 1.2. We show that the minimal covering of a mesh is not approximable in polynomial time within any reasonable bound, unless $\mathcal{P} = \mathcal{NP}$, cf. Theorem 5.2. The subdivision problem is also known to be hard, but the sizes of the instances of the problems occurring in the engineering application are small enough to be solved with standard techniques.

The challenging part of the segmentation lies in the recognition of standard elements in a target geometry. This topic is addressed in the main part of this thesis.

Chapter 2 examines a method for affine shape matching. The main contribution of this chapter is the formal definition of wide bases (Definition 2.6) and a geometric proof (Theorem 2.7) that shows that each compact set in Euclidean space contains a wide base. This enables the analysis of the approximation ratio of an elementary algorithm (Theorem 2.13 and Corollary 2.15) and an extension for a related multi-criteria optimization problem. Section 2.5 describes a method which is tailored to the specific set of transformations which can currently be realized by the manufacturing method developed in the engineering part of the project.

A further extension of this framework can be achieved by expanding the set of transformations which can be applied to standard elements. This would enhance the utility of the process considerably, since even fewer standard elements would be necessary to produce desired target geometries. An interesting approach is related to the problem setting described in Chapter 3. The stretching and shrinking operations which are necessary for forming a target surface are captured by the distortion of a correspondence between the initial and target geometry.

We give several original results on previously open questions on the computational complexity of identifying a correspondence with minimal distortion between two metric spaces. While the results given in Theorems 3.22, 3.24 and 3.28 show that the problem cannot be approximated in polynomial time under standard complexity theoretic assumptions, we introduce methods which provide good results for several classes of instances. Section 3.3 establishes an approach which enables the extension of a relation between few distinct points of the metric spaces — called bases — with low error, see Theorem 3.43. We transfer ideas from convex geometry to identify suitable bases. In conjunction, these methods can be used to obtain a fixed parameter tractable approximation algorithm for the problem, see Theorem 3.47. This yields a promising direction for further analysis and automation of the driving process from a mechanical engineering as well as a mathematical point of view.

We also establish that the non-rigid shape matching problem can be reduced to the dense k-subgraph problem, which asks for an induced subgraph on exactly k vertices of a given graph and is considered in Chapter 4. Similar to the previous chapters, we substantiate that the size of many practical instances can be markedly reduced without significantly deteriorating the objective value of a solution. This results in a (non-polynomial time) $(1 + \frac{1}{k-1})$-approximation algorithm for the problem (Theorem 4.8). We also present a real world application in forestry, in which this approach works particularly well.

Index

Symbols

$\|\cdot\|$, 8
$\|\cdot\|_2$, 8
$|\cdot|$, 8
$\langle\cdot,\cdot\rangle$, 8
2^X, 8
$\mathcal{A}$, 17
$A^\perp$, 8
$\mathbb{B}_x(\varepsilon)$, 9
$D_X(x)$, 57
$E(V)$, 13
$G(V)$, 13
I_d, 8
L_p, 8
$N_G(v)$, 13
T_{B_P,B_Q}, 17
$V(E)$, 13
$X \triangle Y$, 8
$\Pi(X,Y)$, 10
Δ, 95
$\Delta(R,T)$, 16
Δ-core, 96
  weighted, 96–98
$\Gamma(R)$, 40
$\Gamma_p(R)$, 40
$\mathcal{L}_\delta(P,Q)$, 16
$\mathcal{R}(X,Y)$, 10
aff(A), 8
argmax, 8
argmax$_1$, 8
argmin, 8
argmin$_1$, 8
conv(A), 8
diam(X), 9
dim(A), 8
ε-base, 66
par(B), 19, 20
$\mathcal{P}_1(R)$, 10
$\mathcal{P}_2(R)$, 10
$\zeta(R_B)$, 68
d_B, 9
d_h, 9
d_H, 9
d_G, 13
d_{GH}, 39
$d_{GH,p}$, 41
$d_{BGH,p}$, 41
$d_{HI,p}$, 58
$d_{BHI,p}$, 58
p_G, 93

B

base, 17
  flat, 33
  metric space, 66
  wide, 19, 20
    centre, 19

C

correspondence, 10
  bottleneck, 10
  cover, 10

D

distance, 9
  p-Gromov-Hausdorff, 41
  bottleneck, 9
  bottleneck p-Gromov-Hausdorff, 41
  Gromov-Hausdorff, 39, 74
  Hausdorff, 9
  shortest path, 13

E

extreme point, 72
extreme set, 75

G

Gromov, 39

H
Hausdorff, 9
histogram, 56
  distance, 58
    bottleneck, 58
  point, 57

L
largest common point set, 16

M
mesh
  segmentation, 12, 107
metric space, 9
  diameter, 9
  isometry, 9
  rational, 9
minimal covering, 7, 108
minimal dominating set, 108
multi-criteria optimization, 28, 86
multi-terminal cut, 111

P
parallelotope, 19
Pareto front, 28
  approximate, 28
potential, 93
problem
  δ-p-SBGHD, 43
  δ-p-SGHD, 43
  δ-p-SGHD, 54
  p-BGHD, 43, 49, 51, 65
  p-GHD, 43, 50, 64
  DkS, 91
  GHD, 43, 53, 64, 74
  GI, 44
  LCP, 16
  m-equivalence, 44
  MCQ, 54
  MDS, 108
  MI, 43
  MTC, 111
  SAT, 108

R
reduction, 44
  many-one, 44
relation, 10
  bottleneck, 10
  cover, 10
  distortion, 40
  error, 16
  extension, 68
    greedy, 68, 69

S
standard element, 2
subdivision, 7, 110
submesh, 110

T
threshold, 95
transformation, 8
  Euclidean, 8
  flat, 31
  rigid, 8

Bibliography

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. 1st ed. Cambridge University Press, 2009.

[AC09] R. Andersen and K. Chellapilla. "Finding Dense Subgraphs with Size Bounds". In: *Algorithms and Models for the Web-Graph*. Vol. 5427. 2009, pp. 25–37.

[ACG00] C. Ambühl, S. Chakraborty, and B. Gärtner. "Computing Largest Common Point Sets under Approximate Congruence". In: *Algorithms - ESA 2000*. Vol. 1879. 2000, pp. 52–64.

[AFS06] M. Attene, B. Falcidieno, and M. Spagnuolo. "Hierarchical mesh segmentation based on fitting primitives". In: *The Visual Computer* 22.3 (2006), pp. 181–193.

[Aku96] T. Akutsu. "Protein structure alignment using dynamic programing and iterative improvement". In: *IEICE Transactions on Information and Systems* 79.12 (1996), pp. 1629–1636.

[AMCO08] D. Aiger, N. J. Mitra, and D. Cohen-Or. "4-points Congruent Sets for Robust Surface Registration". In: *ACM Transactions on Graphics* 27.3 (2008), pp. 1–10.

[Asa+00] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. "Greedily Finding a Dense Subgraph". In: *Journal of Algorithms* 34.2 (2000), pp. 203–221.

[ATT98] T. Akutsu, H. Tamaki, and T. Tokuyama. "Distribution of Distances and Triangles in a Point Set and Algorithms for Computing the Largest Common Point Sets". In: *Discrete & Computational Geometry* 20.3 (1998), pp. 307–331.

[BB76] H. Barrow and R. Burstall. "Subgraph isomorphism, matching relational structures and maximal cliques". In: *Information Processing Letters* 4.4 (1976), pp. 83–84.

[BBG11] S. Borgwardt, A. Brieden, and P. Gritzmann. "Constrained Minimum-k-Star Clustering and its application to the consolidation of farmland". In: *Operational Research: Special Issue: Optimization in Agriculture (Part II)* 11.1 (2011), pp. 1–17.

[BBI01] D. Burago, I. Burago, and S. Ivanov. *A Course in Metric Geometry*. Vol. 33. American Mathematical Society, 2001.

[BBK06] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. "Efficient computation of isometry-invariant distances between surfaces". In: *SIAM Journal on Scientific Computing* 28.5 (2006), pp. 1812–1836.

[BBK08] A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. 1st ed. Springer, 2008.

[BEP09] A. Billionnet, S. Elloumi, and M.-C. Plateau. "Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method". In: *Discrete Applied Mathematics* 157.6 (2009), pp. 1185–1197.

[BG04] A. Brieden and P. Gritzmann. “A quadratic optimization model for the consolidation of farmland by means of lend-lease agreements”. In: *Operations Research Proceedings 2003: Selected Papers of the International Conference on Operations Research*. 2004, pp. 324–331.

[Bha+10] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. “Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph”. In: *Proceedings of the 42nd ACM Symposium on Theory of Computing*. 2010, pp. 201–210.

[BK07] M. Boutin and G. Kemper. *Lossless Representation of Graphs using Distributions*. Preprint. 2007. URL: `http://arxiv.org/abs/0710.1870`.

[BM92] P. Besl and N. D. McKay. “A method for registration of 3-D shapes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256.

[Bom+99] I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo. “The Maximum Clique Problem”. In: *Handbook of Combinatorial Optimization*. 1999, pp. 1–74.

[Bro+04] A. M. Bronstein, M. M. Bronstein, A. Spira, and R. Kimmel. “Face Recognition from Facial Surface Metric”. In: *Computer Vision – ECCV 2004*. Vol. 3022. 2004, pp. 225–237.

[Bro+10] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro. “A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching”. In: *International Journal of Computer Vision* 89.2–3 (2010), pp. 266–286.

[BS14] S. Borgwardt and F. Schmiedl. “Threshold-based preprocessing for approximating the weighted dense k-subgraph problem”. In: *European Journal of Operational Research* 234.3 (2014), pp. 631–640.

[Bur84] R. E. Burkard. “Quadratic assignment problems”. In: *European Journal of Operational Research* 15.3 (1984), pp. 283–289.

[CB99] S. Chakraborty and S. Biswas. “Approximation Algorithms for 3-D Common Substructure Identification in Drug and Protein Molecules”. In: *Algorithms and Data Structures*. Vol. 1663. 1999, pp. 253–264.

[CGK06] S. Cabello, P. Giannopoulos, and C. Knauer. “On the Parameterized Complexity of d-Dimensional Point Set Pattern Matching”. In: *Parameterized and Exact Computation*. Vol. 4169. 2006, pp. 175–183.

[CH96] J. Chen and Y. Han. “Shortest paths on a polyhedron, Part I: Computing shortest paths”. In: *International Journal of Computational Geometry & Applications* 06.02 (1996), pp. 127–144.

[Cha+95] B. Chazelle, D. P. Dobkin, N. Shouraboura, and A. Tal. “Strategies for Polyhedral Surface Decomposition: An Experimental Study”. In: *Proceedings of the Eleventh Annual Symposium on Computational Geometry*. 1995, pp. 297–305.

[CL06] R. R. Coifman and S. Lafon. “Diffusion maps”. In: *Applied and Computational Harmonic Analysis* 21.1 (2006). Special Issue: Diffusion Maps and Wavelets, pp. 5–30.

[CP84] D. Corneil and Y. Perl. “Clustering and domination in perfect graphs”. In: *Discrete Applied Mathematics* 9.1 (1984), pp. 27–39.

[CSAD04] D. Cohen-Steiner, P. Alliez, and M. Desbrun. “Variational Shape Approximation”. In: *ACM Transactions on Graphics* 23.3 (2004), pp. 905–914.

[CT10] E. Candes and T. Tao. “The Power of Convex Relaxation: Near-Optimal Matrix Completion”. In: *IEEE Transactions on Information Theory* 56.5 (2010), pp. 2053–2080.

[Dah+94] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis. “The Complexity of Multiterminal Cuts”. In: *SIAM Journal on Computing* 23.4 (1994), pp. 864–894.

[Dat05] J. Dattorro. *Convex Optimization & Euclidean Distance Geometry*. Meboo Publishing, 2005.

[DGH98] M. Dyer, P. Gritzmann, and A. Hufnagel. “On The Complexity of Computing Mixed Volumes”. In: *SIAM Journal on Computing* 27.2 (1998), pp. 356–400.

[EH10] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[EI96] A. Efrat and A. Itai. “Improvements on Bottleneck Matching and Related Problems Using Geometry”. In: *Proceedings of the Twelfth Annual Symposium on Computational Geometry*. 1996, pp. 301–310.

[EK01] A. Elad and R. Kimmel. “Bending invariant representations for surfaces”. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. 2001, pp. 168–174.

[FL01] U. Feige and M. Langberg. “Approximation Algorithms for Maximization Problems Arising in Graph Partitioning”. In: *Journal of Algorithms* 41.2 (2001), pp. 174–211.

[FPK01] U. Feige, D. Peleg, and G. Kortsarz. “The Dense k-Subgraph Problem”. In: *Algorithmica* 29.3 (2001), pp. 410–421.

[FS97] U. Feige and M. Seltser. *On the densest k-subgraph problems*. Tech. rep. CS97-16. Weizmann Institute, 1997.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[GK94] P. Gritzmann and V. Klee. “On the complexity of some basic problems in computational convexity: I. Containment problems”. In: *Discrete Mathematics* 136.1–3 (1994), pp. 129–174.

[GMO99] M. T. Goodrich, J. S. B. Mitchell, and M. W. Orletsky. “Approximate geometric pattern matching under rigid motions”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.4 (1999), pp. 371–379.

[Gol09] D. Goldstein. “The dense k-subgraph problem”. MA thesis. The Open University of Israel, 2009.

[Gol84] A. V. Goldberg. *Finding a Maximum Density Subgraph*. Tech. rep. UCB/CSD-84-171. EECS Department, University of California, Berkeley, 1984.

[GP03] C. Grigorescu and N. Petkov. “Distance sets for shape filters and shape recognition”. In: *IEEE Transactions on Image Processing* 12.10 (2003), pp. 1274–1286.

[Gri13] P. Gritzmann. *Grundlagen der Mathematischen Optimierung*. 1st ed. Springer, 2013.

[GRS14] P. Gritzmann, M. Ritter, and F. Schmiedl. “An Approximation for the Largest Common Point Set Problem under Affine Transformations”. 2014. Manuscript.

[GSCO07] R. Gal, A. Shamir, and D. Cohen-Or. “Pose-Oblivious Shape Signature”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.2 (2007), pp. 261–271.

[GT98] B. Günsel and A. M. Tekalp. “Shape similarity matching for query-by-example”. In: *Pattern Recognition* 31.7 (1998), pp. 931–944.

[GWH01] M. Garland, A. Willmott, and P. S. Heckbert. “Hierarchical Face Clustering on Polygonal Surfaces”. In: *Proceedings of the 2001 Symposium on Interactive 3D Graphics*. 2001, pp. 49–58.

[Hal93] M. M. Halldórsson. “Approximating the minimum maximal independence number”. In: *Information Processing Letters* 46.4 (1993), pp. 169–172.

[Hau14] F. Hausdorff. *Grundzüge der Mengenlehre*. Veit and Company, 1914.

[HB01] J. Hightower and G. Borriello. “Location systems for ubiquitous computing”. In: *IEEE Computer* 34.8 (2001), pp. 57–66.

[HFK01] M. Hunting, U. Faigle, and W. Kern. “A Lagrangian relaxation approach to the edge-weighted clique problem”. In: *European Journal of Operational Research* 131.1 (2001), pp. 119–131.

[HHP05] H. Hoffmann, R Hautmann, and R Petry. “Studies for the Development of a Simulation Basis for Numerically Controlled Driving of Sheet Metal”. In: *Advanced Materials Research* 6 (2005), pp. 517–524.

[Hof+10] H. Hoffmann, Z. Yang, D. Scherer, G. Klinker, and F. Echtler. *Entwicklung automatisierter Fertigungsstrategien für das Kopiertreiben*. Tech. rep. 325. European Research Association for Sheet Metal Working, 2010.

[Hop96] H. Hoppe. “Progressive Meshes”. In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. 1996, pp. 99–108.

[HRT97] R. Hassin, S. Rubinstein, and A. Tamir. “Approximation algorithms for maximum dispersion”. In: *Operations Research Letters* 21.3 (1997), pp. 133–137.

[HYZ02] Q. Han, Y. Ye, and J. Zhang. “An improved rounding method and semidefinite programming relaxation for graph partition”. In: *Mathematical Programming* 92.3 (2002), pp. 509–535.

[Hå99] J. Håstad. “Clique is hard to approximate within $n^{1-\varepsilon}$”. In: *Acta Mathematica* 182.1 (1999), pp. 105–142.

[JH99] A. Johnson and M. Hebert. “Using spin images for efficient object recognition in cluttered 3D scenes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.5 (1999), pp. 433–449.

[Joh97] A. E. Johnson. “Spin-images: A representation for 3-D surface matching”. PhD thesis. Carnegie Mellon University, 1997.

[KB91] J. Keil and T. Brecht. “The Complexity of Clustering in Planar Graphs”. In: *The Journal of Combinatorial Mathematics and Combinatorial Computing* 9 (1991), pp. 155–159.

[Kho06] S. Khot. “Ruling Out PTAS for Graph Min-Bisection, Dense k-Subgraph, and Bipartite Clique”. In: *SIAM Journal on Computing* 36.4 (2006), pp. 1025–1071.

[KI93] S. B. Kang and K. Ikeuchi. “The complex EGI: a new representation for 3-D pose determination”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.7 (1993), pp. 707–721.

[KP93] G. Kortsarz and D. Peleg. “On choosing a dense subgraph”. In: *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*. 1993, pp. 692–701.

[KST93] J. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem*. Birkhäuser, 1993.

[KT96] A. Kalvin and R. Taylor. “Superfaces: polygonal mesh simplification with bounded error”. In: *IEEE Computer Graphics and Applications* 16.3 (1996), pp. 64–77.

[KW05] L. Kobbelt and J. Wu. “Structure Recovery via Hybrid Variational Surface Approximation”. In: *Computer Graphics Forum* 24.3 (2005), pp. 277–284.

[Laf04] S. S. Lafon. “Diffusion maps and geometric harmonics”. PhD thesis. Yale University, 2004.

[LDB05] G. Lavoué, F. Dupont, and A. Baskurt. “A new CAD mesh segmentation method, based on curvature tensor analysis”. In: *Computer-Aided Design* 37.10 (2005), pp. 975–987.

[Llo82] S. Lloyd. “Least squares quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137.

[LMZ08] M. Liazi, I. Milis, and V. Zissimopoulos. “A constant approximation algorithm for the densest k-subgraph problem on chordal graphs”. In: *Information Processing Letters* 108.1 (2008), pp. 29–32.

[Mac99] N. MacLeod. “Generalizing and extending the eigenshape method of shape space visualization and analysis”. In: *Paleobiology* 25.1 (1999), pp. 107–138.

[Mau96] C. Maunder. *Algebraic Topology*. Dover Publications, 1996.

[Mit+13] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan. “Symmetry in 3D Geometry: Extraction and Applications”. In: *Computer Graphics Forum* 32.6 (2013), pp. 1–23.

[MMP87] J. Mitchell, D. Mount, and C. Papadimitriou. “The Discrete Geodesic Problem”. In: *SIAM Journal on Computing* 16.4 (1987), pp. 647–668.

[MS00] E. M. Macambira and C. C. de Souza. “The edge-weighted clique problem: Valid inequalities, facets and polyhedral computations”. In: *European Journal of Operational Research* 123.2 (2000), pp. 346–371.

[MS05] F. Mémoli and G. Sapiro. “A Theoretical and Computational Framework for Isometry Invariant Recognition of Point Cloud Data”. In: *Foundations of Computational Mathematics* 5.3 (2005), pp. 313–347.

[MT00] R. D. Monteiro and T. Tsuchiya. “Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions”. In: *Mathematical Programming* 88.1 (2000), pp. 61–83.

[MV98] J. Maintz and M. A. Viergever. “A survey of medical image registration”. In: *Medical Image Analysis* 2.1 (1998), pp. 1–36.

[Mé07] F. Mémoli. “On the use of Gromov-Hausdorff Distances for Shape Comparison”. In: *Symposium on Point Based Graphics*. 2007, pp. 81–90.

[Mé08] F. Mémoli. “Gromov-Hausdorff distances in Euclidean spaces”. In: *Proceedings of the 21st IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2008, pp. 1–8.

[Mé12] F. Mémoli. “Some Properties of Gromov–Hausdorff Distances”. In: *Discrete & Computational Geometry* 48.2 (2012), pp. 416–440.

[Nad89] D. Naddef. “The hirsch conjecture is true for (0, 1)-polytopes”. In: *Mathematical Programming* 45.1–3 (1989), pp. 109–110.

[Nag04] U. Nagendran. *Methods of using wireless geolocation to customize content and delivery of information to wireless communication devices*. US Patent 6,731,940. 2004.

[Nie06] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[Opr+12] D. Opritescu, P. Sachnik, Z. Yang, R. Golle, W. Volk, H. Hoffmann, F. Schmiedl, M. Ritter, and P. Gritzmann. “Automated Driving by Standardizing and Scaling the Manufacturing Strategy”. In: *Procedia CIRP* 3.0 (2012). 45th CIRP Conference on Manufacturing Systems, pp. 138–143.

[OV14] D. Opritescu and W. Volk. “Variation of components by automated driving – A knowledge-based approach for geometric variance”. In: *International Journal of Material Forming* (2014). In press.

[Pat+03] N. Patwari, A. Hero, M. Perkins, N. Correal, and R. O’Dea. “Relative location estimation in wireless sensor networks”. In: *IEEE Transactions on Signal Processing* 51.8 (2003), pp. 2137–2148.

[Pis06] D. Pisinger. “Upper bounds and exact algorithms for p-dispersion problems”. In: *Computers & Operations Research* 33.5 (2006), pp. 1380–1398.

[PS05] C. Papadimitriou and S. Safra. “The Complexity of Low-distortion Embeddings Between Point Sets”. In: *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. 2005, pp. 112–118.

[RDK12] D. Raviv, A. Dubrovina, and R. Kimmel. “Hierarchical Matching of Non-rigid Shapes”. In: *Scale Space and Variational Methods in Computer Vision*. Vol. 6667. 2012, pp. 604–615.

[RS08] M. R. Ruggeri and D. Saupe. “Isometry-invariant Matching of Point Set Surfaces”. In: *Proceedings of the 1st Eurographics Workshop on 3D Object Retrieval*. 2008, pp. 17–24.

[Rus07] R. M. Rustamov. “Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation”. In: *Proceedings of the fifth Eurographics symposium on Geometry processing*. 2007, pp. 225–233.

[Sch03] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Vol. B. Springer, 2003.

[Sch13] A. Schrottenloher. “Surface registration and segmentation with an application in automated metal forming”. MA thesis. Technische Universität München, 2013.

[Sch14a] F. Schmiedl. “A Combinatorial Optimization Approach Non-Rigid Shape Matching”. 2014. Manuscript.

[Sch14b] F. Schmiedl. “The Gromov-Hausdorff Distance and its Application in Non-Rigid Shape Matching”. 2014. Manuscript.

[Sha08] A. Shamir. “A survey on Mesh Segmentation Techniques”. In: *Computer Graphics Forum* 27.6 (2008), pp. 1539–1556.

[SOG09] J. Sun, M. Ovsjanikov, and L. Guibas. “A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion”. In: *Computer Graphics Forum* 28.5 (2009), pp. 1383–1392.

[SS87] J. T. Schwartz and M. Sharir. “Identification of Partially Obscured Objects in Two and Three Dimensions by Matching Noisy Characteristic Curves”. In: *The International Journal of Robotics Research* 6.2 (1987), pp. 29–44.

[STK02] S. Shlafman, A. Tal, and S. Katz. “Metamorphosis of Polyhedral Surfaces using Decomposition”. In: *Computer Graphics Forum* 21.3 (2002), pp. 219–228.

[Str03] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2003.

[Sur+05] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe. “Fast Exact and Approximate Geodesics on Meshes”. In: *ACM Transactions on Graphics* 24.3 (2005), pp. 553–560.

[SWK07] R. Schnabel, R. Wahl, and R. Klein. “Efficient RANSAC for Point-Cloud Shape Detection”. In: *Computer Graphics Forum* 26.2 (2007), pp. 214–226.

[Sø04] M. M. Sørensen. “New facets and a branch-and-cut algorithm for the weighted clique problem”. In: *European Journal of Operational Research* 154.1 (2004), pp. 57–70.

[Tam+13] G. Tam, Z.-Q. Cheng, Y.-K. Lai, F. Langbein, Y. Liu, D. Marshall, R. Martin, X.-F. Sun, and P. Rosin. “Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.7 (2013), pp. 1199–1217.

[Tho98] R. B. Thompson. “Global Positioning System: The Mathematics of GPS Receivers”. In: *Mathematics Magazine* 71.4 (1998), pp. 260–269.

[TKR08] L. Torresani, V. Kolmogorov, and C. Rother. “Feature Correspondence Via Graph Matching: Models and Global Optimization”. In: *Computer Vision – ECCV 2008*. Vol. 5303. 2008, pp. 596–609.

[Vel01] R. C. Veltkamp. “Shape Matching: Similarity Measures and Algorithms”. In: *Shape Modeling International*. 2001, pp. 188–197.

[Vol+13] W. Volk, D. Opritescu, P. Gritzmann, and F. Schmiedl. *Automatisiertes Kopiertreiben – Analyse und Katalogisierung von Bauteilen und Fertigungsstrategien*. Tech. rep. 366. European Research Association for Sheet Metal Working, 2013.

[Wan+12] C. Wang, M. M. Bronstein, A. M. Bronstein, and N. Paragios. "Discrete Minimum Distortion Correspondence Problems for Non-rigid Shape Matching". In: *Scale Space and Variational Methods in Computer Vision*. Vol. 6667. 2012, pp. 580–591.

[Win+11] T. Windheuser, U. Schlickewei, F. Schmidt, and D. Cremers. "Geometrically consistent elastic matching of 3D shapes: A linear programming solution". In: *IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 2134–2141.

[Woh10] K. Wohlgemuth. "Netzsegmentierung für automatisierte Fertigungsverfahren". Diplomarbeit. Technische Universität München, 2010.

[Xia08] M. Xiao. "Algorithms for Multiterminal Cuts". In: *Computer Science – Theory and Applications*. Vol. 5010. 2008, pp. 314–325.

[ZH99] D. Zhang and M. Hebert. "Harmonic maps and their applications in surface matching". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2. 1999, pp. 524–530.

[Ös02] P. R. Östergård. "A fast algorithm for the maximum clique problem". In: *Discrete Applied Mathematics* 120.1–3 (2002), pp. 197–207.

[Łu91] T. Łuczak. "Size and connectivity of the k-core of a random graph". In: *Discrete Mathematics* 91.1 (1991), pp. 61 –68.