Probabilistic Matching of Planar Shapes

Dissertation zur Erlangung des Doktorgrades

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

2009

von

Ludmila Scharf

Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
scharf@mi.fu-berlin.de

gefördert durch die EU im Rahmen des Forschungsprojektes
PROFI, unter Vertragsnummer IST-511572-2
und durch die DFG im Rahmen des Schwerpunktprogramms 1307
Algorithm Engineering

Betreuer: Prof. Dr. Helmut Alt
Institut für Informatik
Freie Universität Berlin
Takustraße 9
D–14195 Berlin
Germany
alt@mi.fu-berlin.de

Gutachter: Prof. Dr. Helmut Alt
Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
Germany
alt@mi.fu-berlin.de

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

Vorlage zur Begutachtung: 13. März 2009
Termin der Disputation: 5. Juni 2009
Fassung vom: 10. Juni 2009

Contents

Abstract

In this thesis we study a probabilistic approach for the shape matching problem. The studied approach is based on an intuitive definition of the shape matching task: Given two shapes A and B find that transformation within the class of allowable transformations which maps B to A in a *best possible way*. A mapping is considered to be good if large parts of the two shapes coincide within some tolerance distance δ.

We assume that the shapes are modeled by finite sets of rectifiable curves in the plane. As possible classes of transformations we consider sub-classes of affine transformations: *translations*, *rigid motions* (translations and rotations), *similarity maps* (translation, rotation, and scaling), *homotheties* (translation and scaling), *shear transformations*, and *affine maps*.

The major idea of the probabilistic algorithm is to take random samples of points from both shapes and give a "vote" for that transformation matching one sample with the other. If that experiment is repeated frequently, we obtain by the votes a certain probability distribution in the space of transformations. Maxima of this distribution indicate which transformations give the best match between the two figures. The matching step of the algorithm is, therefore, a voting scheme.

In this thesis we analyze the similarity measure underlying the algorithm and give rigorous bounds on the runtime (number of experiments) necessary to obtain the optimal match within a certain approximation factor with a prespecified probability. We perform a generic analysis of the algorithm for arbitrary transformation classes, as well as an in-depth analysis for different sub-classes of affine transformations. It is also shown that the density function of the vote distribution is exactly the normalized generalized Radon transform in the cases of translations and rigid motions.

We consider the theoretical analysis as the major contribution of this thesis, since it leads to a better understanding of this kind of heuristic techniques.

Zusammenfassung

In dieser Arbeit wird ein probabilistischer Ansatz zum Vergleichen von Formen untersucht. Der Ansatz entspricht der intuitiven Vorstellung von "Formanpassung": zwei Formen werden als ähnlich empfunden wenn es eine Transformation aus der Menge der erlaubten Transformationen gibt, die die beiden Formen gut zur Deckung bringt. Dabei bedeutet "gute Deckung" dass große Teile einer Form sich in der räumlichen Nähe der anderen Form befinden.

Die Formen, für die wir den Ansatz untersuchen, werden als endliche Mengen von Kurvensegmenten endlicher Länge dargestellt. Als Menge der erlaubten Transformationen betrachten wir Unterklassen der affinen Abbildungen: *Translationen* (Parallelverschiebungen), *starre Bewegungen* (Translationen und Drehungen), *Ähnlichkeitsabbildungen* (Translationen, Drehungen und Skalierungen), *Homothetien* (Translationen und Skalierungen), *Scherungen* und *affine Abbildungen* selbst.

Die Grundidee des Algorithmus ist folgende: Nimm zufällige Punktproben aus jeder der Formen und zeichne eine "Stimme" für die Transformation auf, die die Probe der einen Form auf die Probe der anderen Form abbildet. Nach mehrfacher Wiederholung des Experiments zeichnet sich im Transformationsraum eine gewisse Verteilung der Stimmen aus. Die Maxima dieser Verteilung deuten "gute" Transformationen an, wobei eine gute Transformation eine ist, die große Teile der Formen zur Deckung bringt.

In dieser Arbeit untersuchen wir das dem Algorithmus zu Grunde liegende Ähnlichkeitsmaß und bestimmen die notwendige Anzahl an Experimenten um eine gute Anpassung von Formen innerhalb einer vorgegebenen Approximationsschranke mit vorgegebener Erfolgswahrscheinlichkeit zu bestimmen. Der durchgeführte Analyse ist generisch und gilt für beliebige Transformationsklassen. Zusätzlich führen wir eine ausführliche Analyse für die oben genannten Transformationsklassen durch. Weiterhin zeigen wir, dass die vom Experiment induzierte Wahrscheinlichkeitsdichtefunktion im Transformationsraum genau der verallgemeinerten Radon-Transformation für Translationen und starre Bewegungen entspricht.

Die theoretische Analyse betrachten wir als den Hauptbeitrag dieser Arbeit, denn sie führt zum besseren Verständnis einer Klasse von Heuristiken die unter dem Sammelnamen "Abstimmungsmethoden" bekannt ist.

Acknowledgments

I would like to thank all the people that have supported me in writing this thesis. First of all, I would like to thank my advisor, Helmut Alt, for excellent guidance and intensive research, for many good ideas and discussions, for being very understanding and supporting, and for the careful proofreading of this thesis. I thank Remco Veltkamp for agreeing to co-referee this thesis and for doing a great job as the project coordinator of the PROFI-project. I am grateful to all participants of the PROFI-project for the excellent research environment and for the interesting discussions. Many thanks go to my co-authors for sharing their knowledge and experience with me, and for many fruitful and interesting discussions: Helmut Alt, Stefan Felsner, Elena Mumford, Marc Scherfenberg, Sven Scholz, Daria Schymura. Of course I thank all members of the work group "Theoretical Computer Science" at Freie Universität Berlin for their generous help and support, for many many fruitful discussions, for the friendly atmosphere and excellent research environment, and for the great coffee rounds. I am grateful to my family and all my friends for their friendship and their support and for having so much good time together. Finally, and most importantly, I thank my husband Leo much more than I can express within these lines.

Berlin, March 2009.

Chapter 1

Introduction

Shape Matching

Our research is motivated by the task of automated retrieval of figurative images in large databases, where the evaluation of the similarity of two images is based on their geometric shape and not color or texture. Matching two geometric shapes under transformations and evaluating their similarity is one of the central problems in such retrieval systems, but it is also a problem of independent interest which is widely covered in literature, as surveys by Alt and Guibas [3], Hagedoorn and Veltkamp [34], and Latecki and Veltkamp [41] indicate.

We assume that the shapes are modeled by finite sets of rectifiable curves in the plane. As possible classes of transformations we will consider *translations*, *rigid motions* (rotation and translation) and *similarities* (rotation, scaling and translation) and more general *affine transformations*. Our objective is to develop an algorithm which allows an efficient implementation and whose result comes close to human similarity perception.

Several similarity measures and algorithms are known to match two curves, especially polygonal curves. One of the most widely used similarity measures is the Hausdorff distance which is defined for any two compact sets A and B. Alt et al. describe in [3, 4] efficient algorithms for computing the Hausdorff distance and minimizing it under translations and rigid motions for arbitrary sets of line segments. One of the drawbacks of the Hausdorff distance is that it is very sensitive to noise, i.e., parts of the input data that result from some errors in a digitalization process. A few similarity measures are defined for pairs of curves, which capture the relative course of two curves: the Fréchet distance by Alt and Godau [2], the turning function distance by Arkin et al. [8], and the dynamic time warping distance by Efrat et al. [24]. There are only a few generalizations of those distances to sets of curves: Alt et al. [5] define a generalization of the Fréchet distance to geometric graphs, and Tanase et al. [56] describe an algorithm for matching a set of polygonal curves to a single polygon. Further, Dumitrescu and Rote [21] define the Fréchet distance of a set of curves. However, they consider a slightly different problem, namely, measur-

ing the similarity within one set of curves, whereas in the shape matching task we consider two sets of curves and want to determine the similarity between them. One similarity measure which is designed for comparing sets of curves is the reflection visibility distance described by Hagedoorn et al. [35]. This measure is robust against different kinds of disturbances in the input data but is expensive to compute.

There is a vast variety of shape matching methods developed in the image processing and computer vision community which do not require shapes to consist of a single curve. A large group of matching algorithms can be generally described as follows: In a preprocessing phase a signature vector is computed for each shape. During the matching phase the similarity of the shapes is determined by the similarity of the corresponding signature vectors. A signature vector can be a representation of the shape by Fourier descriptors, as in [15, 27, 39], or wavelet descriptors [14, 17, 47]. A signature can also be an aggregation of some statistical features of the complete shape, e.g., aspect ratio, circularity, and relative area, as well as features of the parts of the boundary, e.g., right-angleness or sharpness, see [22]. Two such signatures are then compared component-by-component and a similarity value for the shapes is aggregated from the similarity of the individual components of the signature vectors.

Another type of signature vectors is constructed by selecting some points along the shape, e.g., equally spaced points, and for each of these points computing a set of characteristic values. The characteristic values of a point can be, for example, the curvature and the length of the adjacent segments [53], the integral of the curvature function over a certain interval of the curve containing that point [18], the curvature and the distances to all other points of the shape [28], the curvature and the distance to the centroid [40]. The optimal alignment for two such vectors is found in the first three cited articles ([18, 28, 53]) using dynamic programming techniques that are similar to string alignment algorithms. In [40] a good alignment is found using a technique called particle filters, which maintains an alignment-probability matrix, where the matrix element p_{ij} denotes the probability of the i-th element of the first signature vector to match the j-th element of the second. Thus, every possible correspondence between an element of one signature vector and an element of the other is considered simultaneously and is represented by a probability value in the alignment-probability matrix. The probabilities are iteratively re-computed and after a certain number of iteration steps the alignment with the highest probability is considered to be the best one. A similar idea can be found in matching methods based on expectation maximization (EM) algorithms (see, e.g., [43, 58]). However, EM algorithms consider just one possible matching transformation, or one set of model fitting parameters in each iteration step. Observe, that the signature vectors consisting of point features are suitable only for shapes composed of a single contour curve, since the order of the points in the signature vector is relevant for the alignment step.

Another class of shape matching algorithms is known under the general name "voting schemes". It includes geometric hashing [59], alignment methods [37, 49], the

generalized Hough transform, also called pose clustering [1, 55], the Radon transform [31, 57], and the RANSAC algorithm by Fischler and Bolles [26]. A more detailed description of these methods is included in Section 2.3.1.

Common to many of the methods described above is the representation of the shapes by pixel images or by a finite set of points, which might be connected to form polygonal curves. In the latter case the points or polygonal curves are usually the result of an edge detection process in pixel images. Clearly, pixel images are the typical source of the data for computer vision applications. Consequently, the definition of the similarity measures or of the point features for the algorithms working with polygonal curves relies on the fact that the vertices of a polygon are almost equally spaced along the curve and are close to each other. Such assumptions are not necessary for the algorithm presented in this work.

The method we introduce is close to an intuitive notion of "matching", i.e., find one or more candidates for the best transformations, which when applied to the shape B map the most similar parts of the two shapes to each other. The major idea is to take random samples of points from both shapes and give a "vote" for that transformation (translation, rigid motion, or similarity) matching one sample with the other. If that experiment is repeated frequently, we obtain by the votes a certain probability distribution in the space of transformations. Maxima of this distribution indicate which transformations give the best match between the two figures. The matching step of our algorithm is, therefore, a voting scheme. The idea of random sampling for geometric problems with an analysis similar to ours has been used in a more general context by Cheong, Efrat, and Har-Peled [16]. Further, Mitra, Guibas, and Pauly [44] use a related random sampling technique, but different clustering methods, for symmetry detection in 3D geometric models.

Related methods in the image processing community are the generalized Hough transform and the RANSAC algorithm. In contrast to those methods we do not consider a discrete set of features that describe shapes, but work with continuous curves. Our method is independent of the choice of the parameterization and the discretization grid in the transformation space. In addition, we give rigorous bounds on the runtime (number of experiments) necessary to obtain the optimal match within a certain approximation factor with a prespecified probability. This work gives a generic analysis of the algorithm for arbitrary transformation classes, as well as an in-depth analysis for different sub-classes of affine transformations. We also show that the density function of votes is exactly the normalized generalized Radon transform in case of translations and rigid motions. The classical definition and analysis of Radon transform can be found in Gel'fand et al. [29]. Toft [57] formulates a definition of generalized Radon transform in the context of image analysis, and Ginkel et al. [30] give a comparative analysis of the generalized Hough and Radon transforms. For the reader's convenience we include the definition of the generalized Radon transform in Section 2.3.3.

We consider the theoretical analysis as the major contribution of this work, since

it leads to a better understanding of this kind of heuristic techniques. In fact, our algorithm is not meant to be directly applied to shape comparison problems arising in practice. For such purposes it makes sense to modify our technique and enhance it with heuristic methods, which we did (see Alt et al. [6]) within a shape retrieval system developed in the EU-funded project PROFI. Its major application, in cooperation with the industrial PROFI-partner Thomson-Compumark in Antwerp, is to identify (illegal) similarities between new trademark designs and existing trademarks of various companies in a large trademark database.

Overview of the Thesis

In this thesis we investigate a probabilistic approach to matching two planar shapes modeled by sets of rectifiable curves under several classes of transformations, such as translations, rigid motions and similarity transformations.

First, in Chapter 2 we discuss preliminaries that will be used in several of the following chapters and give an overview of the related methods in the computational geometry and in the image processing community. In Chapter 3 we describe the proposed probabilistic matching algorithm in a generic way and analyse the underlying similarity measure. We derive rigorous bounds on the number of experiments required by the algorithm in order to achieve an approximation of the underlying similarity measure within the given error bounds with high probability. In Section 3.3 we provide the details of the algorithm for different subclasses of affine transformations and discuss alternative variants of the algorithm for some transformation classes. In Section 3.4 we analyse the approximation of the similarity measure by a grid counting method. Further, we discuss the robustness of the algorithm against different types of disturbances in shape representation and some possible generalizations of the presented method to matching higher-dimensional shapes. In Chapter 4 we show that the density function of the probability distribution induced by the algorithm in the transformation space corresponds to the generalized Radon Transform for the case of translations and for rigid motions. We also analyze the distribution of the scaling factor for a modified version of the random experiment. The latter analysis is performed only for some simple types of shapes.

Chapter 2

Preliminaries

In this preliminary chapter we collect some known concepts and techniques which are used several times along this thesis. Definitions of general concepts, such as, e.g., curve or transformation, are restricted to a two-dimensional case since the subject of this work is matching of shapes in the plane.

2.1 Representation of Shapes

In general there are many ways to describe a geometric shape. The representation varies depending on the application and the field of research. In the most general case a geometric shape is a compound object which consists of a finite number of basic objects, such as points, straight line segments, smooth curves or bounded regions, as illustrated in Figure 2.1. Another representation of shapes which is commonly used in

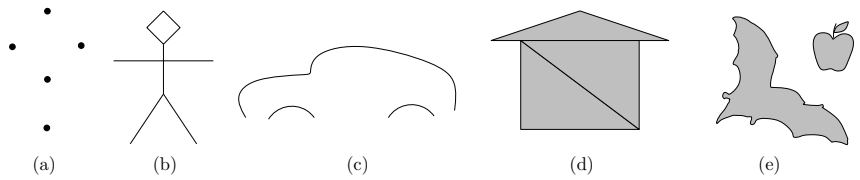


Figure 2.1: Examples of geometric shape representations by points, straight line segments, curve segments and regions.

the field of computer vision and pattern recognition is by a set of pixels. These can be seen as special sets of points, which are arranged on a grid. However, representation by higher dimensional objects, such as lines and regions, allows a more compact description. One can describe a shape either as a solid volume, or by its boundary. In two dimensions this yields the special case of closed curves representing the boundary of a shape. Curves are a class of shapes which are worthwhile to consider since they appear in many applications. Furthermore, psychological experiments indicate that for human perception the shape information is concentrated along the boundary,

see for example Feldman and Singh [25], Rogowitz and Voss [52]. Therefore, it is reasonable to consider shapes that are curves for the purpose of similarity evaluation.

Throughout this thesis we consider shapes represented by sets of rectifiable curves in the plane, which includes sets of straight line segments, boundary curves of solid objects, and shapes composed of smooth curves. Thus, in Figure 2.1 shapes (b) and (c) and the boundaries of regions in shapes (d) and (e) are in scope of this work.

We now describe in detail the representation of shapes considered in this thesis.

Definition 2.1 (Curve/Polygonal Curve)**.** A *curve* in a plane is the image of a continuous mapping $\mathcal{C} : I \to \mathbb{R}^2$ for a closed interval $I = [a, b] \subset \mathbb{R}$. If additionally $\mathcal{C}(a) = \mathcal{C}(b)$ then the curve $\mathcal{C}$ is said to be *closed*.

A plane *polygonal curve* P of complexity $n \in \mathbb{N}$ is a curve $P : [0, n] \to R^2$ with vertices $P_i := P(i)$ for $i \in \{0, 1, 2, \ldots, n\}$ and line segments $\overline{P}(j) := P|_{[j,j+1]}$ for $j \in \{0, 1, \ldots, n-1\}$, such that $P(j + \lambda) = (1 - \lambda)P(j) + \lambda P(j + 1)$ for all $\lambda \in [0, 1]$.

The representation of a curve by a mapping is called *parametrization* of a curve. We call a bijective continuous mapping $\alpha : I' \to I$ from a closed interval I' into a closed interval I, such that the inverse of α is also continuous, a *reparameterization*, since for a curve $\mathcal{C} : I \to \mathbb{R}^2$ it defines a reparameterized curve $\mathcal{C} \circ \alpha : I' \to \mathbb{R}^2$ that has the same image points in $\mathbb{R}^2$.

If $\mathcal{C} : I \to \mathbb{R}^2$ represents a smooth curve, then we denote by $\mathcal{C}'(r)$ the differential displacement vector along $\mathcal{C}$ at the point $\mathcal{C}(r)$. A parameter form representation $\mathcal{C} : I \to \mathbb{R}^2$ is called a *constant speed parameterization* if for all $r \in I$ the length of the vector $\mathcal{C}'(r)$ is constant. We will assume that the shapes are represented by a set of piecewise smooth curves and that for each smooth piece a constant speed parameterization is available. Obviously, this is the case for polygonal curves, which would be the most common representation in practice.

Definition 2.2. The *length* L of a smooth curve $\mathcal{C} : I \to \mathbb{R}^2$ is defined as

$$L = \int_{r \in I} \left|\mathcal{C}'(r)\right| \mathrm{d}\, r \ ,$$

where $|\cdot|$ denotes the length of a vector.

The length of a curve is invariant under reparametrizations and is therefore a geometric property of a curve. A curve is said to be *rectifiable* if its length is finite. Thus, we only consider curves of finite length.

A parameterization $\mathcal{C}$ is called a *unit speed parameterization* if $|\mathcal{C}'(r)| = 1$ for all $r \in I$. It is also known as *natural parameterization* or *parameterization by length* since in this case the length of the parameter interval I corresponds to the length of the curve, $I = [0, L]$, and for every parameter value $r \in I$ it holds that $r = \int_0^r |\mathcal{C}'(r)| \,\mathrm{d}\, r$.

Under the assumption that a curve is given in a natural or constant speed parametrization, which can be computed in constant time for every parameter value, selecting a random point on the curve under uniform distribution with respect to

length is equivalent to selecting randomly a real number in a given interval. Assuming that a random number can be generated in constant time, selection of a random point can be performed in constant time.

2.2 Geometric Transformations

The transformation classes considered in this work are sub-classes of affine transformations. Although the algorithm presented in this work is very general and can be applied to a broader class of transformations, we do not provide the details for projective or other non-linear transformations. This decision is motivated by the fact that the most commonly used transformations for the shape matching task are translations, rigid motions, and similarities.

Affine transformations: An *affine transformation* in the plane is a map $t : \mathbb{R}^2 \to \mathbb{R}^2$ of the form $t(p) = M \cdot p + v$, where

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$$

is a linear transformation matrix and $v = (v_x, v_y) \in \mathbb{R}^2$ is a translation vector. An affine transformation preserves collinearity, i.e., three collinear points are still collinear after the transformation, and ratios of distances, e.g., the midpoint of a line segment remains the midpoint after transformation.

An affine map has six parameters, so each map t can be associated with a point $(m_1, m_2, m_3, m_4, v_x, v_y) \in \mathbb{R}^6$ and vice versa. In order to exclude degenerate transformations, i.e., non-invertible transformations, we restrict $\mathbb{R}^6$ to the points where the corresponding linear transformation matrix M has a non-zero determinant. The subset of $\mathbb{R}^6$ defined as $\big\{(m_1, m_2, m_3, m_4, v_x, v_y) \in \mathbb{R}^6 \mid m_1 m_4 - m_2 m_3 \neq 0\big\}$, where every point represents an affine transformation map, is called the *space of affine transformations*. We denote the space of transformations by $\mathcal{T}$. In the remainder of this work we use the same symbol for a transformation map $t : \mathbb{R}^2 \to \mathbb{R}^2$ and for the corresponding point $t \in \mathcal{T}$.

Given two ordered sets of three non-collinear points in the plane (a_1, a_2, a_3) and (b_1, b_2, b_3) there exists a unique affine transformation t such that $t(b_i) = a_i$ for $i \in \{1, 2, 3\}$. The parameters, or coordinates, of t are the solution of the system of six linear equations of the form $M \cdot b_i + v = a_i$. Three is also the smallest size of point sets such that an affine transformation mapping one set to the other is uniquely defined. For sets consisting of four or more points there is in general no affine transformation mapping exactly the points of one set to those of the other. Therefore we say that the *minimum sample size* for affine transformations is three.

Translations: The first sub-class of affine transformations, that we will consider, is the class of translations. A translation is a map $t : \mathbb{R}^2 \to \mathbb{R}^2$ of the form $t(p) =$

$p + v$, where $v \in \mathbb{R}^2$ is the translation vector. The linear transformation matrix is the identity matrix and is therefore omitted in the formula. Thus, there are two parameters describing a translation map, and the space of translations is a two-dimensional subspace of the space of affine transformations. Translation is a length, angle and orientation preserving transformation.

For two points a and b there exists a unique translation that maps b to a. Therefore, the minimum size of a sample needed to determine a translation map is one.

Rigid motions: The next class we consider is the class of rigid motions. A rigid motion is a combination of rotation around the origin by an angle α and translation by a vector $v \in \mathbb{R}^2$. A linear transformation matrix M of a rigid motion has the form

$$M = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} ,$$

that is, $m_1 = m_4 = \cos\alpha$ and $m_2 = -m_3 = -\sin\alpha$. There are three parameters describing a rigid motion, and therefore the space of rigid motions is three-dimensional. Rigid motions preserve lengths, angles and orientations.

There are two commonly used possibilities to parameterize the space of rigid motions. The intuitive way is to use the rotation angle and coordinates of the translation vector. In this case we have to perform computations with trigonometric functions. Another possibility is to use the matrix and translation vector parameters (m_1, m_2, v_x, v_y) to describe a rigid motion. Thus, a rigid motion is represented by a four-dimensional point. Since not every four-dimensional point represents a valid rigid motions, we need to restrict the four-dimensional space to a three-dimensional algebraic variety by the constraint $m_1^2 + m_2^2 = 1$. We will use the latter representation for computational reasons in this work, but will refer to the parameterization by angle and translation vector in order to provide an intuitive description.

Observe that there is no point sample that uniquely determines a rigid motion: for two points a and b and for every rotation angle α there exists a translation vector v_α such that the rigid motion with parameters (α, v_α) maps b to a. Thus, two samples consisting each of one point are not sufficient to determine uniquely a rigid motion. If we have two ordered sets consisting of two (or more) points each, then, in general, there is no rigid motion mapping each point of one set to the corresponding point of the other. In this case we say that rigid motion is overdetermined by samples of two or more points.

If we would somehow associate with every point a direction information then a rigid motion mapping one point to the other in such a way that their directions coincide would be uniquely determined. We discuss later different variants of assigning direction information to a point of a shape, as well as different approaches of determining a transformation from small samples of two shapes.

Similarity transformations: Another commonly used class of transformations is the class of similarity maps, which combines rotation around the origin, scaling, and translation. A similarity transformation is a map $t : \mathbb{R}^2 \to \mathbb{R}^2$ of the form $t(p) = M \cdot p + v$, where

$$M = \begin{pmatrix} m1 & m_2 \\ -m_2 & m_1 \end{pmatrix}$$

is the linear transformation matrix with non-zero determinant and v is the translation vector. Rotation by angle α and scaling by factor k is expressed by parameters of the matrix M, where $m_1 = k \cdot \cos\alpha$ and $m_2 = -k \cdot \sin\alpha$. A similarity transformation preserves angles and the ratio of lengths of the segments.

The space of similarity maps is four dimensional and can be parameterized by rotation angle, scaling factor, and coordinates of the translation vector, or by the parameters of the linear transformation matrix and the coordinates of the translation vector. We choose the latter parameterization for computational reasons.

For any two ordered sets of two points (a_1, a_2) and (b_1, b_2) there exists a unique similarity transformation that maps b_1 to a_1 and b_2 to a_2. Therefore, we say that the minimum size of a sample needed to determine a similarity map is two.

Homothetic transformations: A homothety is a map $t : \mathbb{R}^2 \to \mathbb{R}^2$ of the form $t(p) = k \cdot p + v$, where k is the scaling factor and v is the translation vector. Thus, a homothety combines scaling and translation. The parameters of the linear transformation matrix are $m_1 = m_4 = k$ and $m_2 = m_3 = 0$. The space of homothetic transformations is three dimensional. Homothetic transformations preserve angles and the ratio of lengths of the segments.

Like in the case of rigid motions, there is no point sample that uniquely determines a homothetic transformation. We discuss in Section 3.3.4 different approaches to determine a homothetic transformation using small samples of shapes.

Shear transformations: A shear is an affine transformation in which all points along a given line l remain fixed while other points are shifted parallel to l by a distance proportional to their perpendicular distance from l, as illustrated in Figure 2.2. A shear is an area preserving transformation.

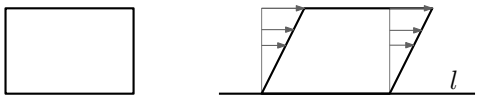


Figure 2.2: Shear transformation along a line l.

Commonly considered are horizontal shear transformations, or shears parallel to

the x-axis, which are defined by a linear transformation matrix

$$M_{\text{hor.shear}} = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix}$$

and map the unit vector $(0, 1)$ to the vector $(m, 1)$, and vertical shear transformations, or shears parallel to the y-axis, defined by a matrix

$$M_{\text{vert.shear}} = \begin{pmatrix} 1 & 0 \\ m & 1 \end{pmatrix} \quad .$$

The parameter m is the negative tangent of the declination angle φ of a vertical line, $m = -\tan\varphi$, in case of a horizontal shear, and $m = \tan\varphi$ for a vertical shear, where φ is the inclination angle of a horizontal line, as depicted in Figure 2.3. Thus, the shear along the x-axis with parameter m maps a vertical line $x = a$ to a line $y = (x - a)/m$ of a slope $1/m$ for $m \neq 0$, and with $m = 0$ we have an identity transformation.

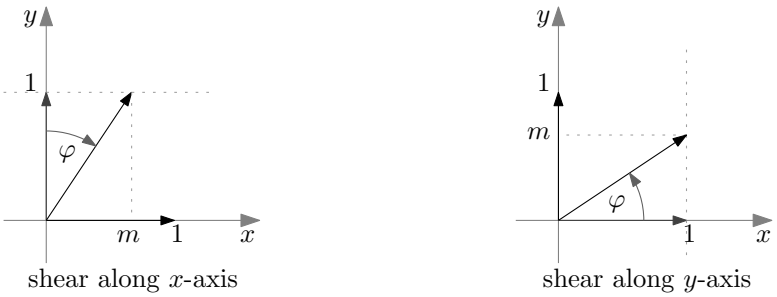


Figure 2.3: Horizontal and vertical shear.

The space of shear transformations is three dimensional and can be parameterized by (m, v_x, v_y).

For two points a and b there is no unique transformation that maps b to a, rather for every shear parameter m there exists a unique translation vector that completes the desired transformation. For a pair of two points (a_1, a_2) and (b_1, b_2), in general, there exists no shear transformation that maps b_1 to a_1 and b_2 to a_2. Different approaches to computing a shear transformation from a small sample of two shapes are discussed in Section 3.3.5.

2.3 Related Work

2.3.1 Voting Schemes

In this section we briefly describe related methods in the image processing and pattern recognition community, which can be combined under the common name "voting

schemes". These methods include RANSAC, geometric hashing, alignment methods, and the generalized Hough or Radon transform, also known as pose clustering.

Common to all these methods is the representation of shapes by a finite set of so-called feature points. The point set representing a shape might be a set of characteristic points extracted from a pixel image, a set of points of high curvature extracted from a curve or a set of curves, or some set of points that are important or characteristic to the shape according to some criterion.

The problem addressed by these methods is the following: Given an observed shape A, a model shape B and the class of allowed transformations, determine whether A is an instance of B and if so, find parameters of the transformation that when applied to B results in a maximum correspondence between A and B. Let m denote the number of points in A and n the number of points in B.

RANSAC. RANSAC is an abbreviation for **Ra**ndom **Sa**mple **C**onsensus, an iterative algorithm by Fischler and Bolles [26]. Apart from the pattern matching application, RANSAC is also used in statistical data analysis for estimating parameters of a mathematical model from a set of observed data. The main assumption made by the algorithm is that the data consist of *inliers*, that is, the points in A that correspond to some point in B, and *outliers*, the points in A that have no correspondences in B.

In one iteration the algorithm draws a random sample S_A of A consisting of the minimum number of points, say k, needed to uniquely determine a transformation of the given class, as described in the previous section. Then it computes for every ordered k-tuple S_B of points in B the size of the so-called *consensus set*, the set of points in A that have a counterpart in B when the transformation mapping S_B to S_A is applied to B. Hereby a point $a \in A$ is considered to be an inlier if there is a point in the transformed shape B with distance at most some error tolerance δ from a.

The algorithm terminates after some number of iterations N. The result is either the size of the largest consensus set together with the corresponding transformation parameters, or the assertion that there is no "large enough" consensus set. In both cases the result is correct with some probability η. A consensus set is said to be large enough, if the ratio of inliers is not smaller than a threshold value ρ.

Consider a sample S_A drawn in one iteration step. Assuming that the ratio of inliers in A is at least ρ, the probability that S_A does not consist of k inliers is at most $1-\rho^k$. Observe that once the algorithm selects a sample consisting of k points of the largest consensus set, it finds the corresponding transformation and the size of the largest consensus set in the deterministic step of the iteration. Therefore, the result is wrong if the algorithm fails to select a sample consisting of k inliers in each of N iteration steps. The failure probability is, therefore, $(1-\rho^k)^N$. If this probability has to be bound by $1-\eta$, the number of iterations should be $N = \lceil \log_{1-\rho^k}(1-\eta) \rceil$. Thus, the total complexity of the algorithm is $O(N \cdot n^k \cdot T_C)$, where T_C is the time

needed to compute the size of the consensus set.

Finally, if the ratio of inliers is not known in advance and should be discovered by the algorithm, the number of iterations N is initialized to some large value. Every time a larger consensus set is found N is adjusted according to the new inlier ratio.

Geometric hashing. Geometric hashing is a two-stage object recognition algorithm, see Wolfson and Rigoutsos [59] for an overview article.

In the preprocessing step coordinates of the model points are computed with respect to all reference frames defined by the points of the model shape B. These coordinates are then stored in a hash table. A reference frame, also called basis, is a set of points defining the coordinate system. The type and size of a reference frame depends on the class of the allowed transformations. So for example, for matching under translations a single point suffices as basis, since it determines the origin of the coordinate system and the orientation of coordinate axes is fixed. For rigid motions a reference frame consists of two points, where the first point defines the origin of the coordinate system and the second the orientation of the x-axis. Thus, if a reference frame for the given class of transformations consists of k points, then for *every* ordered k-tuple of points of the model shape B the coordinates of all points of B are computed with respect to that k-tuple as the basis.

In the recognition phase a similar procedure is applied to the observed shape A, that is, for every k-tuple of points in A the coordinates of all points are computed with respect to that k-tuple as the reference frame. Then a pair of reference frames for the model and observed shapes gets a vote for each matched point coordinates pair. The reference frame pairs with high number of votes are the candidate solutions. For every candidate solution the model is transformed according to basis information and the match is verified.

Geometric hashing requires $O(n^{k+1})$ additional storage space for each model shape, where n is the number of points in the model shape. The preprocessing time is $O(n^{k+1})$, and the recognition time is $O(m^{k+1}T_H)$, where m is the number of points in the observed shape and T_H is the time needed to access the hash table.

Alignment methods. The basic idea of the alignment method is to determine hypothesized alignments, or transformations, from small sets of observed shape points and model points, see Huttenlocher and Ullman [37], Olson [49]. These alignments can then be tested to determine if they are correct.

The general idea of an alignment method is to iteratively choose a basis set S_B of the model shape B and a basis set S_A of observed shape A. A basis set is a set of points of minimum size k needed to uniquely determine a transformation of a given class. The transformation t mapping S_B to S_A is computed and verified, that is, t is applied to B and the number of points in both shapes that are mapped into a certain proximity of each other is calculated. The basis pair with the highest number of correspondences determines the best match.

The choice of the basis sets can be performed deterministically or randomly. In deterministic case the time complexity of the algorithm is $O(n^k m^k T_V)$, where n and m denote the size of model and observed shapes, respectively, k is the size of a basis set, and T_V is the verification time.

In case of random choice, the algorithm terminates if with high probability either the best match is found or it can be stated that the model is not present in the observed shape. The randomized variant of the alignment method is closely related to the RANSAC algorithm.

The generalized Hough transform. The ideas of alignment methods have motivated development of other techniques cited under different names such as cluster methods, pose clustering, evidence gathering, hypothesis accumulation, the generalized Hough transform, or the generalized discrete Radon transform. The articles by Aguado et al. [1], Ginkel et al. [31], Olson [48], Stockman [55], Toft [57] are just a small portion of the vast variety of literature on these methods. Instead of performing a verification process for each transformation of the model, the set of all possible transformations is considered as a cluster space according to a suitable parametrization. The usual clustering techniques used together with pose clustering are binning and multidimensional histogramming.

The transformation space, or cluster space is considered to be discretized by some grid, and every grid cell is used as an accumulator for the "votes". Then for every point a of the observed shape and every point b of the model shape a vote is added to every accumulator cell containing a transformation mapping b to a. Since in general the region of all possible transformations mapping the point b to the point a is unbounded, the possible transformations are usually limited to some predefined region. Alternatively, a vote is casted for every minimum size subset of points of the model and observed shape that uniquely define a transformation of the given class mapping one subset to the other.

After all point pairs of the two shapes or pairs of minimum size subsets have been considered, the number of votes in each cluster cell of the transformation space is considered to be the measure of consistency for a transformation representing that cell, e.g., center of the cell. Some authors add an additional verification procedure for the transformations with the maximum number of votes. In the verification step the candidate transformation is applied to the model shape B and the number of correspondences between the observed shape A and the transformed copy of B is counted.

Discussion. While sharing some similarities, our method is fundamentally different from the described pattern recognition methods. There is a conceptual difference in the representation of shapes: We do not require any discretization or feature points identification, but work with continuous sets of points corresponding to planar curves. Further, we avoid the costly verification procedure performed by RANSAC and align-

ment methods for every sample pair. In fact, by defining the similarity measure that can be directly computed from the clustering information, we avoid completely the verification step. At the same time our method requires no additional storage, in contrast to geometric hashing. Finally, we do not restrict the transformation space to a predefined region. Our method is independent of the choice of parameterization and discretization grid in transformation space.

2.3.2 Related Work in Computational Geometry

Cheong, Efrat, and Har-Peled [16] introduced a probabilistic framework based on the theory of ε-approximations. The basic idea behind it is that the area of a planar region can be estimated by counting the number of points of an ε-approximation inside the region. The ε-approximations are generated by random sampling. The authors apply the described framework to the problems of maximizing the visibility region inside a simple polygon, maximizing the Voronoi region, and the problem of maximizing the area of overlap of two simple planar polygons under translations and rigid motions.

For the shape matching problem it is shown that using a uniform random sample the maximum area of overlap of two polygons under translations can be approximated with an absolute error at most ε and failure probability at most $1/n^6$ in time $O(m + (n^2/\varepsilon^4)\log^2 n)$, where n and m are the complexities of the polygonal shapes. Although the problem considered in this work does not involve area computation or maximization we use related ideas in the argumentation of the approximation bounds proofs.

Mitra, Guibas, and Pauly [44] use a related random sampling technique for symmetry detection in 3D geometric models. There two point samples are taken from one shape and a transformation mapping one sample to the other is recorded in transformation space. Then clusters of "votes" in transformation space are detected using a mean shift clustering technique.

2.3.3 The Generalized Radon Transform

In this section we briefly describe the Radon transform and a generalized definition of it. This definition is used in Chapter 4 to show that the density function underlying the random experiment in our algorithm is exactly the generalized Radon transform of one shape with respect to the other.

The Radon transform in two dimensions is named after Johann Radon who showed in [51] that a function $f : \mathbb{R}^2 \to \mathbb{R}$ can be represented by values of its integral along straight lines. Since every straight line can be uniquely represented by two parameters, Radon transform is a mapping from the function into the two dimensional parameter space of straight lines. The inverse Radon transform corresponds to the reconstruction of the function from the integral projections.

Consider the normal form, also called Hesse standard form, of a line g:

$$g : \tau - x\cos\theta - y\sin\theta = 0 \ , \tag{2.1}$$

where θ is the angle of inclination of the normal to g and τ is the distance from the origin to g. The corresponding parametric form of g is

$$\begin{aligned} g_x(s) &= \tau\cos\theta - s\sin\theta \\ g_y(s) &= \tau\sin\theta + s\cos\theta \end{aligned} \tag{2.2}$$

for $s \in \mathbb{R}$. The Radon transform of f with respect to the parameterized family of lines is then

$$\begin{aligned} \mathcal{R}_f(\tau,\theta) &= \int_{s\in\mathbb{R}} f(\tau\cos\theta - s\sin\theta, \tau\sin\theta + s\cos\theta)\,\mathrm{d}\,s \\ &= \int_{x\in\mathbb{R}}\int_{y\in\mathbb{R}} f(x,y)\delta(\tau - x\cos\theta - y\sin\theta)\,\mathrm{d}\,y\,\mathrm{d}\,x \ , \end{aligned}$$

where $\delta(\cdot)$ is the Dirac delta-function. Mathematical properties of the Radon transform have been extensively studied in [29].

Ginkel et al. [30] further generalized the definition of the Radon transform in the context of shape detection. They consider the task of detecting a k-dimensional shape in a D-dimensional image, which is also considered to be a collection of k dimensional objects. More precisely, given an image, that is, a set of k dimensional objects in D dimensions, and a family of k-dimensional shapes (also in D dimensions) that can be described by d parameters, the task is to find the parameters corresponding to the best fitting member of the family of shapes.

For example, the image could be a set of planar curves, that is, a set of one-dimensional objects ($k = 1$) in two dimensions ($D = 2$). A parameterized family of shapes could be given by an arbitrary planar curve and the set of allowed transformations of that curve. Then the number of parameters d describing the family of shapes is the dimension of the transformation space, and the parameters of a shape instance are the parameters of the corresponding transformation.

First we introduce notation as used in [30]:

c denotes the model shape.

c^t denotes the model shape c with parameters t, or with transformation t applied.

$c(s)$ is an internal parametrization of a model shape c by parameter s, that is, coordinates of a point of shape c corresponding to the parameter value s (e.g., parametric form of a straight line).

$c^t(s)$ denotes the coordinates of a point $c(s)$ transformed by t.

$\mathcal{C}(p,t)$ is a set of $D-k$ constraint functions that define a shape, where p is a point in $\mathbb{R}^D$ and t is a transformation. All constraint functions $\mathcal{C}_i(p,t)$ with $1 \le i \le D-k$ evaluate to zero iff the point p lies on the shape c transformed by t.

$I(p)$ denotes a D-dimensional image that consists of k-dimensional shapes. The image is also given in constraint form: $I(p) = \prod_{j=1}^{D-k} \delta(\mathcal{C}_j^I(p))$, where $\mathcal{C}_j^I(p) = 0$ for all $1 \leq j \leq D-k$ iff p is a point of an image shape.

Observe that every set $\mathcal{C}(p)$ of $D-k$ constraint functions has to fulfill the property that a D-dimensional integral of $\prod_{i=1}^{D-k} \delta(\mathcal{C}_i(p))$ over a region $R \subset \mathbb{R}^D$ evaluates to the k-dimensional volume of the part of the shape contained in R.

The generalized Radon transform of an image I with respect to the parameterized family of shapes $c(t)$ can then be formulated in the following equivalent ways:

$$\begin{aligned} \mathcal{R}_{\boldsymbol{c}}(I,t) &= \int_{p \in c^t} I(p)\,\mathrm{d}\,p \\ &= \int_{s \in \mathbb{R}^k} I(c^t(s)) \left\| \frac{\partial c}{\partial s} \right\| \mathrm{d}\,s \qquad (2.3) \\ &= \int_{p \in \mathbb{R}^D} I(p) \prod_{i=1}^{D-k} \delta\left(\mathcal{C}_i(p,t)\right) \mathrm{d}\,p \ . \qquad (2.4) \end{aligned}$$

The generalized Radon transform provides a mapping from the image space to the parameter (transformation) space. The function created in transformation space contains peaks for those t, for which c^t is present in the image. If there is a shape c^{t^*} in the image I and $t \neq t^*$ then $\mathcal{R}_{\boldsymbol{c}}(I,t)$ evaluates to a finite number proportional to the number of intersections between c^t and c^{t*}. In case $t = t^*$, $\mathcal{R}_{\boldsymbol{c}}(I,t)$ yields a large response proportional to the k-dimensional volume of the shape c^{t*}.

The shapes considered in this work are planar curves, that is, one-dimensional objects in two-dimensional image space. An algebraic curve in the plane can be descried by one constraint equation. The polygonal curves can be viewed as a collection of straight line segments. For each segment the parametric representation is exactly the parametric representation given by Equation (2.2) of the line containing that segment with parameter s restricted to some interval. Similarly, a constraint function for a line segment is the constraint function of the line containing that segment, Equation (2.1), with additional restrictions on point coordinates. This constraint function fulfills the integral conditions mentioned above.

Chapter 3

Probabilistic Matching

In this chapter we describe the probabilistic shape matching algorithm in a generic way, and analyze what distance measure between shapes is being minimized and how many experiments are necessary to get reasonable results. Further, we discuss the details and the properties of the algorithm for individual transformation classes.

3.1 The Probabilistic Algorithm

We assume that the shapes are modeled by finite sets of rectifiable curves in the plane as described in Section 2.1, whereas we do not take into account the direction of curves given by the parameterization, i.e., the shape consists of all points lying on the union of curves. We also assume that for each curve a random point under uniform distribution can be generated in constant time. This is the case for line segments, which would be the most common representation in practice, but also for other curves for which a natural parameterization is explicitly given.

Given two shapes $A, B \subset \mathbb{R}^2$, a class of allowed transformations $\mathcal{T}$ and a certain parameter $\delta > 0$, we want to find a transformation $t \in \mathcal{T}$ which lets the transformed image of B, $t(B)$, *match best* A within a tolerance of δ. We assume that the distance metric for the points in the image space is an algebraic function, for example a general L_p metric. Commonly used distance measures are Euclidean distance (L_2), Manhattan distance (L_1), or maximum distance (L_∞). A *δ-neighborhood* of a point p is defined as $U_\delta(p) = \{x \in \mathbb{R}^2 \mid \text{dist}(x, p) \leq \delta\}$, where $\text{dist}(x, p)$ is the distance with respect to the chosen metric.

The analysis of the similarity measure between shapes underlying this algorithm and thus a definition of what exactly a good match means is given in Section 3.2 for the general case and in more details for different transformation classes in Section 3.3. Here we follow an intuitive notion of a "good match": two shapes are similar if they can be mapped to each other in such a way that large parts of them are close, this position is then a good match. So we are searching for a transformation that maps the most similar parts of the shapes A and B to each other.

The idea of the *probabilistic approach* is quite simple. We first describe an algo-

rithm for matching under translations:

1. Take a random point a from the shape A and a random point b from B and give one "vote" to the translation t which maps b to a, that is $t = a - b$.

2. Repeat this experiment many times. Then the distribution of votes in the two-dimensional translation space $\mathcal{T}$ approximates a certain probability distribution.

3. For a prespecified neighborhood size δ return the points of $\mathcal{T}$ with the highest number of votes in their δ-neighborhood as candidates for good transformations.

The idea behind this algorithm is that the transformations, that map large parts of shapes to each other, should get significantly more votes than others. The size of the δ-neighborhood influences the quality of the match.

For more complex classes of transformations two points are not sufficient to determine a unique transformation, therefore, several points or a point and a direction vector might be necessary to form a *random sample* in the first step of the algorithm. The size of a random sample within one experiment and the shape and the size of the δ-neighborhoods depend on the class of transformations allowed as described in Section 2.2. Further, we denote by $S_B \xrightarrow[\delta]{t} S_A$ the fact that the transformation t maps every element of S_B into the δ-neighborhood of the corresponding element of S_A. A "vote" generated by a pair of random samples S_A and S_B is a *δ-region* in the transformation space, which is defined as the set of transformations t such that $S_B \xrightarrow[\delta]{t} S_A$. For transformation classes other than translations the shape of a δ-region depends on the sample pair generating it.

Before giving a generic variant of the algorithm we briefly describe the random samples and δ-regions for the basic transformation classes:

Translations: For translations, as described above, a random sample consists of a single randomly selected point of each shape, $a \in A$ and $b \in B$, since two points determine uniquely a translation mapping one point to the other. The δ-neighborhood of a translation vector t in the two-dimensional transformation space is defined as a set of vectors that have distance at most δ to vector t, where distance is measured with respect to the metric chosen in image space.

Rigid motions: In case of rigid motions the transformation space is three dimensional. A rigid motion $t = (\alpha, v_x, v_y)$ is defined by a rotation angle α and a translation $v = (v_x, v_y)$. For computational reasons it is more convenient to work with the four-dimensional parameterization (m_1, m_2, v_x, v_y) where $m_1 = \cos\alpha$ and $m_2 = -\sin\alpha$ and an additional constraint $m_1^2 + m^2 = 1$ as described in Section 2.2. Which means that we consider the space of rigid motions as a three dimensional variety in the four dimensional space of similarity transformations. For a general L_p

metric the δ-regions are then bounded by algebraic surfaces, and for the L_1 and L_∞ metrics a region is a convex polytope. We are then interested in the arrangement induced by those four-dimensional regions on the three-dimensional variety described by $m_1^2 + m^2 - 1 = 0$. For the remainder of this section we will use the intuitive parameterization of the space of rigid motions and give the details about using the four dimensional embedding in Section 3.3.2.

We will consider two approaches for randomized matching under rigid motions.

Approach 1: We use a single random point of each shape $a \in A$ and $b \in B$ as a sample in one random experiment and record a δ-region in the space of rigid motions as the set of transformations that map the point b into the δ-neighborhood of point a, while all rotation angles are allowed.

In the three dimensional space parameterized by the rotation angle and translation vector the δ-region corresponding to a sample pair (a, b) in this approach has the shape of a spiral tube extending from 0 to 2π in the direction of the rotation axis, where for each value $\alpha \in [0, 2\pi]$ the cross-section parallel to the translation plane has the shape of the δ-neighborhood with respect to the chosen metric in image space. The structure of δ-regions is described in more detail in Section 3.3.2 and an illustration is provided in Figure 3.5.

Approach 2: A random sample of a shape within one experiment contains random points $a \in A$ and $b \in B$ and for each of the points an angle θ_a and θ_b, respectively, defined by the direction of the tangent line at this point, i.e., $S_A = (a, \theta_a)$, $S_B = (b, \theta_b)$. Two such point-angle pairs define uniquely a rigid motion $t = (\alpha, v_x, v_y)$, such that $\theta_b + \alpha = \theta_a$ and $t(b) = a$. Here we have a special case where a sample consists of different types of data: a point and an angle. So it is reasonable to have different values for the tolerance bounds: $\delta = (\delta_1, \delta_2)$, where δ_1 defines a neighborhood of points, and δ_2 restricts the maximal allowed difference of directions. Thus, a δ-region corresponding to a sample pair is the set of rigid motions t' such that the angle between the tangent at b after rotation and the tangent at a is at most δ_2 and the distance between $t'(b)$ and a is at most δ_1.

A δ-region corresponding to a sample pair $((a, \theta_a), (b, \theta_b))$ is a part of the spiral tube defined by the points a, b and the distance tolerance value δ_1 which is bounded by the planes $\alpha = \theta_a - \theta_b + \delta_2$ and $\alpha = \theta_a - \theta_b - \delta_2$.

We also experimented with yet other approaches for rigid motions but the two we present here give good results and fit into the general framework of our analysis.

Similarity transformations: For similarity maps the transformation space is four-dimensional. A similarity map $t = (\alpha, k, v_x, v_y)$ is defined by a rotation angle α, scaling factor k and a translation vector $v = (v_x, v_y)$. t maps a point $b \in \mathbb{R}^2$ to a point $t(b) = Mb + v$, where

$$M = \begin{pmatrix} k\cos\alpha & -k\sin\alpha \\ k\sin\alpha & k\cos\alpha \end{pmatrix} = \begin{pmatrix} m_1 & m_2 \\ -m_2 & m_1 \end{pmatrix}.$$

A random sample from the shapes contains two points $S_A = (a_1, a_2)$ of A, and two points $S_B = (b_1, b_2)$ of B, which determine a unique similarity transformation t mapping b_1 to a_1 and b_2 to a_2. Although a standard way to parameterize the space of similarity transformations is by (α, k, v_x, v_y), for computational reasons it is more convenient to use the parameterization (m_1, m_2, v_x, v_y) where $m_1 = k\cos\alpha$ and $m_2 = -k\sin\alpha$. For a general L_p metric a δ-region is then bounded by algebraic surfaces, and for the L_1 and L_∞ metrics it is a convex polytope bounded by four pairs of parallel hyperplanes.

Affine transformation: A more general class of transformations are the affine maps. An affine transformation $t = (M, v)$, where

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$$

is a linear transformation matrix and $v = (v_x, v_y)$ is a translation vector, is defined by six parameters. The transformation space is six-dimensional in this case. An affine transformation t maps a point $b \in \mathbb{R}^2$ to a point $Mb + v$. Three non-collinear points in each shape $a_1, a_2, a_3 \in A$ and $b_1, b_2, b_3 \in B$ determine a unique affine transformation that maps b_i to a_i, $i \in \{1, 2, 3\}$. Therefore, a random sample taken in step one of the algorithm consists of three points of each shape.

Generic probabilistic algorithm: Now we can describe the probabilistic algorithm in a generic way: Given two shapes A and B, a class of allowed transformations $\mathcal{T}$ and a certain tolerance parameter δ, we want to find a transformation $t \in \mathcal{T}$ which lets $t(B)$ in some sense match best A within a range of δ:

1. Take random samples S_A from A and S_B from B of an appropriate size and record the δ-region corresponding to this sample pair.
2. Repeat this experiment many times, say N.
3. Take the points of $\mathcal{T}$ covered by the highest number of δ-regions as candidates for good transformations.

In the next section we analyze the probability distribution in transformation space induced by the algorithm and provide the bounds on the number of experiments needed to approximate the maximum of this distribution within a certain factor with a prespecified probability.

3.2 Generic Analysis

3.2.1 Hit Probability in Transformation Space

First we introduce some formal notation and definitions. Let Ω denote the sample space, i.e., the set of all sample pairs (S_A, S_B). By the definition of our random exper-

iment, the samples of two figures are drawn independently and uniformly, therefore, we have a uniform distribution on Ω, i.e., for any subset $R \subset \Omega$ the probability of a randomly drawn sample to be in R is $\Pr(R) = \frac{|R|}{|\Omega|}$, where $|\cdot|$ denotes the Lebesgue measure.

Let $\mathcal{T} \subset \mathbb{R}^d$ denote the d-dimensional transformation space. Consider for every transformation vector t the random variable $Y : \mathcal{T} \times \Omega \to \{0, 1\}$ defined as

$$Y(t, (S_A, S_B)) = \begin{cases} 1 & \text{if } S_B \xrightarrow[\delta]{t} S_A, \\ 0 & \text{otherwise.} \end{cases}$$

For a transformation t and a sample pair (S_A, S_B) Y indicates whether S_A and S_B match under t, that is, whether t maps every component of S_B into a δ-neighborhood of the corresponding component of S_A. That means, the pre-image of 1 of the random variable Y for a fixed transformation t is the set

$$M_\delta(t) = \left\{ (S_A, S_B) \in \Omega \mid S_B \xrightarrow[\delta]{t} S_A \right\} .$$

Therefore, the probability that within one random experiment the value of $Y(t, \cdot)$ is 1 is $\Pr(Y(t, \cdot) = 1) = \frac{|M_\delta(t)|}{|\Omega|}$ which we denote by $p_\delta(t)$ – the hit probability function in transformation space. We formalize the above observation in the following remark:

Remark 3.1. *The hit probability $p_\delta(t)$ in the transformation space induced by the generic algorithm described in Section 3.1 has its maximum at the transformation maximizing the Lebesgue measure of the set $M_\delta(t)$ defined as*

$$M_\delta(t) = \left\{ (S_A, S_B) \in \Omega \mid S_B \xrightarrow[\delta]{t} S_A \right\} .$$

We can interpret the Lebesgue measure of the set $M_\delta(t)$ as a similarity measure associated with a transformation t. Intuitively, this should reflect the perceived notion of "closeness" of two shapes.

The role of the parameter δ. In the description of the algorithm we introduced a parameter δ, which defines how far apart two samples are allowed to be that are still considered as being close. The choice of δ, therefore, controls the trade-off between the quality of match and the size of the parts matched. With a small value of δ our algorithm would find a transformation which maps nearly congruent parts of two shapes to each other. A large value of δ leads to a transformation which gives a rough match for larger parts of the shapes, see Figure 3.1.

The value of δ does not determine whether the matching is partial or complete, in fact the algorithm always performs a partial matching in a sense that the corresponding parts of two shapes are mapped to each other. Neither does δ determine how large the matched parts are, it only specifies how exact the match should be. For nearly congruent figures a small neighborhood size already leads to a complete-complete matching, see Figure 3.2(a). If figure B is nearly congruent to some parts

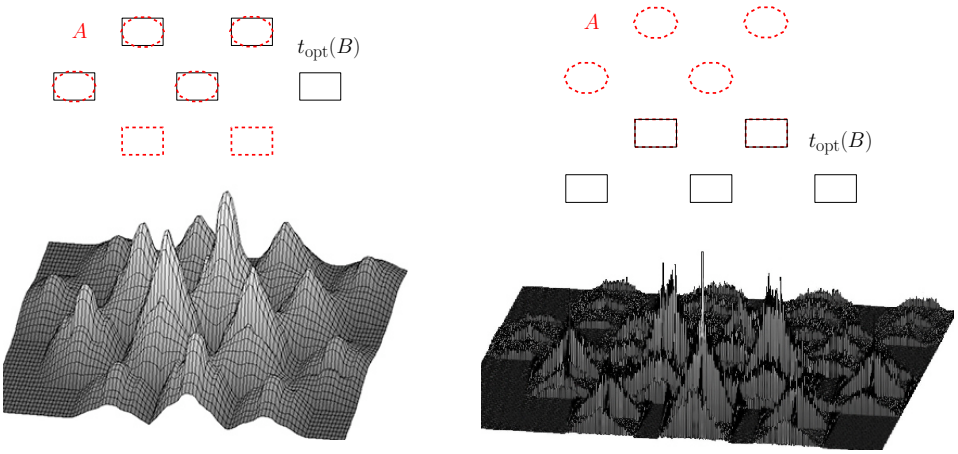


Figure 3.1: Matching under translations with large (left) and small (right) values of δ and the corresponding probability distributions in translation space.

of A, then still with a small value of δ we detect the occurrences of B in A, that is, find a complete-partial matching as shown in Figures 3.2(b) and 3.2(c). The problem of partial-partial matching is not uniquely defined since there is a certain correlation between the quality of match and the size of the matched parts. We address this problem by letting the user specify the quality of match through the choice of δ, for which we then find the matching parts.

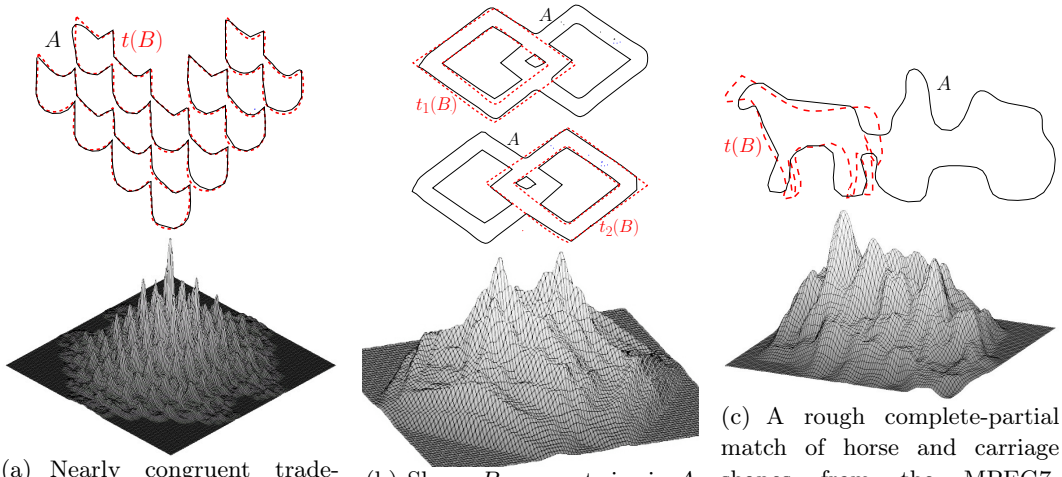
For some applications it might be worth to consider several local maxima of the distribution, since they can give us additional information about the shapes. For example, multiple local maxima of the distribution, that are almost equally good, indicate multiple occurrences of one shape, or its parts, depending on the value of the similarity measure, within the other, see Figure 3.2(b).

3.2.2 Rating the Shape Similarity

We showed above that the objective function underlying the probabilistic algorithm is the hit probability $p_\delta(t)$, which is exactly the measure of the set $M_\delta(t)$ normalized by the measure of the sample space Ω. Informally, the measure of the set $M_\delta(t)$, and thus also the hit probability $p_\delta(t)$, indicates the *amount of correspondence* between two shapes A and B for the transformation t. This measure alone does not tell us whether two given shapes are similar to each other or not. Since we consider the maximum of that similarity or closeness evidence over all possible transformations, we know how much correspondence we can possibly get for the two shapes.

For the matching scenario where we have a shape A and a set of shapes $\mathcal{B} = \{B_1, \ldots, B_k\}$ and want to determine the shapes in $\mathcal{B}$ that are most similar to A, which is a typical scenario for a retrieval task, it might be sufficient to order the shapes of the set $\mathcal{B}$ according to their maximum hit probability. Then the shapes with higher maximum hit probability are those with higher correspondence amount.

Note that the hit probability is in this case a better similarity criterion than the



(a) Nearly congruent trademark images

(b) Shape B occurs twice in A

(c) A rough complete-partial match of horse and carriage shapes from the MPEG7-Shape B data set

Figure 3.2: Matched shapes and the corresponding probability distributions in translation space

measure of the set M_δ: Consider an example where the shape B_1 is a translated copy of the shape A and the shape B_2 contains every curve of the shape A twice with a very small offset ($\ll \delta$), as illustrated in Figure 3.3. Assume that the class of allowed transformations is the class of translations. Let L_A, L_{B_1}, L_{B_2} denote the total lengths of the shapes A, B_1, B_2, respectively, with $L_{B_1} = L_A$ and $L_{B_2} = 2L_A$. The maximum measure of M_δ for the shape B_1 is approximately δL_A, since for the translation mapping B_1 to A for every point a of A there is a part of a curve in B_1 of length approximately δ, such that every point of that part is in the δ-neighborhood of a. The maximum measure of M_δ for the shape B_2 is approximately $2\delta L_A$, since the contours of A are doubled in B_2. Thus, according to the maximum measure of M_δ the shape B_2 is rated to be much more similar to A than its exact copy. The maximum hit probability p_δ for the shape B_1 is approximately $\frac{\delta L_A}{L_A L_{B_1}} = \frac{\delta}{L_{B_1}} = \frac{\delta}{L_A}$ and for the shape B_2 it is $\frac{2\delta L_A}{L_A L_{B_2}} = \frac{2\delta}{L_{B_2}} = \frac{\delta}{L_A}$. Ranking by maximum value p_δ rates both shapes equally similar to A, which is more intuitive.

Another possibility is to normalize the amount of correspondence measured between two shapes by the correspondence of a shape to itself. Again we have a choice of normalizing the maximum measure of M_δ or the maximum value of p_δ. Let for now μ_{AB} denote the maximum measure of similarity over all possible transformations for two shapes A and B. The measure μ_{AB} can either be p_δ or $|M_\delta|$. Further, let μ_A and μ_B denote the maximum of the corresponding similarity measure for the shape A compared to itself and for the shape B compared to itself, respectively. Then the ratio $\frac{\mu_{AB}}{\mu_A}$ measures the amount of correspondence of the shape B relative to A, that is, how well B covers A, and $\frac{\mu_{AB}}{\mu_B}$ the amount of correspondence of A to B. Then for a complete-partial matching for A and B i.e., for a question how

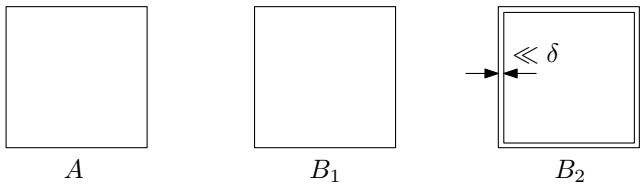


Figure 3.3: An example of shapes A, B_1, B_2, for which the maximum of the measure of the set $M_\delta(t)$ for shapes A and B_2 is approximately twice the maximum measure of the set $M_\delta(t)$ for shapes A and B_1, whereas the maximum values of $p_\delta(t)$ are approximately equal for both pairs of shapes.

well does the complete shape A match to possibly some part of B, we are interested in the measure $\frac{\mu_{AB}}{\mu_A}$. For a complete-complete matching the appropriate measure is the $\min\left\{\frac{\mu_{AB}}{\mu_A}, \frac{\mu_{AB}}{\mu_B}\right\}$. Note that for the scenario of comparing a shape A to a set of shapes $\mathcal{B}$ with respect to complete-partial matching the ranking of the shapes in $\mathcal{B}$ by μ_{AB_i} is equivalent to the ranking by the relative measure.

Which of the measures M_δ or p_δ is more appropriate as a normalized similarity measure depends on the application, we discuss the differences for the above example of the shapes A, B_1, B_2. Let us first consider the maximum of M_δ as the similarity measure, that is $\mu_{AB_1} = \max_t M_\delta(t)$ for the shapes A and B_1 and let μ_{AB_2}, μ_A, μ_{B_1}, μ_{B_2} denote the corresponding maximum measures of M_δ. As mentioned above, $\mu_{AB_1} = \mu_A = \mu_{B_1} \approx \delta L_A$, $\mu_{AB_2} = 2\delta L_A$, and $\mu_{B_2} \approx 2\delta L_{B_2} = 4\delta L_A$. The normalized measures for the shapes A, B_1 are then $\frac{\mu_{AB_1}}{\mu_A} = \frac{\mu_{AB_1}}{\mu_{B_1}} = 1$, which indicates that the shapes A and B_1 have a 100% correspondence (within a tolerance value δ). For the shapes A and B_2 the measure normalized relative to A $\frac{\mu_{AB_2}}{\mu_A} \approx 2$ indicates that the shape B_2 covers the shape A twice, which is exactly the case here. Normalizing with respect to B_2 $\frac{\mu_{AB_2}}{\mu_{B_2}} \approx 1/2$ indicates that A covers only a half of B_2. Although this is true, since A is only half as large as B_2, in many cases (e.g. for rating visual resemblance) we would want A to be still rated as similar to B_2. For the maximum of the hit probability p_δ we get for all four normalized similarity measures the value 1, which means that A is very similar to B_1 and B_1 is very similar to A, but also that A is very similar to B_2 and B_2 is very similar to A. The latter rating reflects better the visually perceived similarity, although the first one captures better the mathematical relations of the shape measures. Since this research is motivated by comparing shapes according to their visual similarity we favor measuring similarity by relative hit probability.

3.2.3 Approximation of the Hit Probability by Arrangement

In this section we determine how many samples are needed in order to approximate the hit probability function $p_\delta(t)$ in the transformation space within a certain accuracy ε with high probability and analyze the total running time of the algorithm.

In order to find a transformation covered by the highest number of δ-regions corresponding to the samples, we consider the arrangement of these δ-regions. All transformations in the same cell of the arrangement have the same region coverage. Therefore, it is sufficient to traverse the arrangement and take the nodes with the highest number of δ-regions that contain this node.

We first show that the number of δ-regions covering the deepest cell of the arrangement gives a good approximation to the maximum value of the hit probability. The number of necessary samples is expressed in terms of the allowed approximation error ε and the maximal probability of failure η. The running time of the algorithm depends on the time needed to determine the deepest cell in the arrangement, which grows exponentially with the dimension of the space, the detailed analysis of running time is given below. We also show that an approximation of the depth of the arrangement as described by Aronov and Har-Peled in [9] results in an approximation of the maximum value of the hit probability with an error of at most 2ε. The speed-up in determining the deepest cell of the arrangement results directly in a speed-up of the probabilistic matching algorithm.

Let the random variable $Z(t)$ denote the sum of the N independent random variables $Y(t,(S_{A1},S_{B1}))$, $\ldots, Y(t,(S_{A\,N},S_{B\,N}))$, as defined in Section 3.2.1, corresponding to the set of N samples in our algorithm. $Z(t)$ counts the number of δ-regions produced by N random experiments that cover t. Note that the expected value of $Z(t)$ is $\mathrm{E}(Z(t)) = p_\delta(t)N$. In the algorithm we find a transformation which is contained in the highest number of δ-regions of the sample set, that is, a transformation $t \in \mathcal{T}$ that maximizes the value of $Z(t)$. Let $\tilde{p}_\delta(t)$ denote the ratio of the number of the observed δ-regions that cover t to the total number of samples, that is $\tilde{p}_\delta(t) = \frac{Z(t)}{N}$. $\tilde{p}_\delta(t)$ is an estimate of $p_\delta(t)$.

Using Chernoff bounds, see Mitzenmacher and Upfal [45], and the technique described by Cheong et al. in [16] we can bound the relative error for the estimate of the hit probability in the transformation space.

The following theorem bounds the number of samples needed for an approximation with a relative error at most ε:

Theorem 3.1. *Given two shapes A and B modeled by finite sets of n rectifiable curves with total lengths L_A, L_B respectively and a tolerance value $\delta > 0$, for any parameter values ε, η, $0 < \varepsilon, \eta < 1$, the following holds: Let t_{app} be a transformation maximizing $\tilde{p}_\delta(t)$ and t_{opt} a transformation maximizing $p_\delta(t)$, and let ν, $0 < \nu < 1$ be a parameter such that either $p_\delta(t_{\mathrm{opt}}) > \nu$, or otherwise we say that the shape A and B do not match well. Then there exists a constant c such that after*

$$N \geq c\frac{1}{\varepsilon^2\nu}\ln\left(\max\left(\frac{1}{\eta}, \frac{1}{\varepsilon^2\nu}\right)\right)$$

random experiments the following holds:

(i) If $p_\delta(t_{\mathrm{opt}}) > \nu$ the probability that the estimate of the maximum of $p_\delta(t)$ has a

relative error larger than ε is at most η, that is

$$\Pr\left[\ |\tilde{p}_\delta(t_{\text{app}}) - p_\delta(t_{\text{opt}})| \geq \varepsilon p_\delta(t_{\text{opt}})\ \right] \leq \eta\ .$$

(ii) If $p_\delta(t_{\text{opt}}) \leq \nu$ the probability that $\tilde{p}_\delta(t_{\text{app}}) \geq (1+\varepsilon)\nu$ is at most η.

For translations, rigid motions (Approach 1) and similarity transformations it holds:

(iii) There exists a constant c such that the probability that $|\tilde{p}_\delta(t_{\text{app}}) - p_\delta(t_{\text{opt}})| \geq \varepsilon p_\delta(t_{\text{opt}})$ is at most η after

$$N \geq c\frac{m^2}{\varepsilon^2\delta^2}\ln\left(\max\left(\frac{1}{\eta}, \frac{m^2}{\varepsilon^2\delta^2}\right)\right)$$

random experiments, where $m = \max(L_A, L_B, n\delta)$.

That means that the number of required experiments in order to get an ε-approximation with probability at least $1-\eta$ is in $O\left(\frac{1}{\varepsilon^2\nu}\ln\left(\max\left(\frac{1}{\eta}, \frac{1}{\varepsilon^2\nu}\right)\right)\right)$ for general transformations, where the parameter ν is a lower bound for the value of the similarity function which will be approximated within guaranteed error bounds. In case that the maximum value of the similarity function is below ν we can say that the shapes do not match well. Parameter ν can be eliminated for the transformation classes for which we can a priori determine a lower bound for the maximum of the similarity function. Such lower bound can be found for translations, rigid motions and similarity maps. The required number of experiments is then bounded by $O\left(\frac{m^2}{\varepsilon^2\delta^2}\ln\left(\max\left(\frac{1}{\eta}, \frac{m^2}{\varepsilon^2\delta^2}\right)\right)\right)$.

Observe that the relative error with respect to $p_\delta(t)$ is also the relative error with respect to $|M_\delta(t)|$, which is the similarity measure underlying our algorithm.

Proof of the relative error bound

We first show that for any transformation vector t and a given threshold ν with high probability we either get a good approximation of $p_\delta(t)$, if this value is at least ν, or otherwise make sure that we do not overestimate it, where with high probability means that the probability of a large error falls exponentially with the number of experiments N.

Lemma 3.1. *For all $0 < \varepsilon, \nu < 1$ for a sample S of size N and any transformation vector $t \in \mathbb{R}^d$ the following holds:*

- $p_\delta(t) \leq \nu \Rightarrow \Pr(\tilde{p}_\delta(t) > (1+\varepsilon)\nu) \leq e^{-\frac{\varepsilon^2\nu N}{3}}$
- $p_\delta(t) \geq \nu \Rightarrow \Pr(|\tilde{p}_\delta(t) - p_\delta(t)| > \varepsilon p_\delta(t)) \leq 2e^{-\frac{\varepsilon^2\nu N}{4}}$.

Proof. First we show that if $p_\delta(t) \leq \nu$ the probability of overestimating this value falls exponentially with N:

$$\begin{aligned}
\Pr(\tilde{p}_\delta(t) > (1+\varepsilon)\nu) &= \Pr(Z(t) > (1+\varepsilon)\nu N) = \Pr(e^{rZ(t)} \geq e^{r(1+\varepsilon)\nu N}) \quad \text{for all } r > 0 \\
&\leq \frac{\mathrm{E}(e^{rZ(t)})}{e^{r(1+\varepsilon)\nu N}} \qquad \text{by the Markov inequality [45, Theorem 3.1]} \\
&\leq \frac{e^{(e^r-1)\,\mathrm{E}(Z(t))}}{e^{r(1+\varepsilon)\nu N}} \qquad \text{see [45, Section 4.2.1]} \\
&= \frac{e^{(e^r-1)p_\delta(t)N}}{e^{r(1+\varepsilon)\nu N}} \qquad \text{since } \mathrm{E}(Z(t)) = p_\delta(t)N \\
&\leq \left(\frac{e^{(e^r-1)}}{e^{r(1+\varepsilon)}}\right)^{\nu N} \qquad \text{since } p_\delta(t) \leq \nu \\
&= \left(e^{\varepsilon-(1+\varepsilon)\ln(1+\varepsilon)}\right)^{\nu N} \qquad \text{for } r = \ln(1+\varepsilon) \\
&\leq e^{-\frac{\varepsilon^2\nu N}{3}} \qquad \text{for } 0 < \varepsilon < 1 \ .
\end{aligned}$$

In case $p_\delta(t) \geq \nu$:

$$\begin{aligned}
\Pr(|\tilde{p}_\delta(t) - p_\delta(t)| > \varepsilon p_\delta(t)) &= \Pr(|Z(t) - p_\delta(t)N| > \varepsilon p_\delta(t)N) \\
&= \Pr(|Z(t) - \mathrm{E}(Z(t))| > \varepsilon\,\mathrm{E}(Z(t))) \\
&\leq e^{-\frac{\varepsilon^2\,\mathrm{E}(Z(t))}{2}} + e^{-\frac{\varepsilon^2\,\mathrm{E}(Z(t))}{4}} \\
&\text{by the simplified Chernoff bound [45, Thm. 4.4,4.5]} \\
&\leq 2e^{-\frac{\varepsilon^2 p_\delta(t)N}{4}} \\
&\leq 2e^{-\frac{\varepsilon^2\nu N}{4}} \qquad \text{since } p_\delta(t) \geq \nu,
\end{aligned}$$

which concludes the proof. □

We associate with each cell C of the arrangement $\mathcal{A}$ of δ-regions a so-called *witness point* p, i.e., a point that lies on a lowest-dimensional face F of $\mathcal{A}$ that contributes to the boundary of C. Observe that F must be completely contained in the boundary of C and is in general a connected component of the intersection of k boundaries of δ-regions with $1 \leq k \leq d$. The dimension of F is $d - k$. Thus, by considering all k-tuples of δ-regions and taking a point in each connected component of their intersection we can be sure to have at least one witness point for each cell of the arrangement.

Since we assume that the distance metric is an algebraic function of constant degree, every intersection of k δ-region boundaries has a constant number of conected components. If δ-regions are convex, which is the case for translations and for similarities in combination with L_1 and L_∞ distance metric, each intersection of k boundaries has just one connected component. Thus, the total number of witness points is at most $c\sum_{k=1}^{d}\binom{N}{k} \leq cN^d$, where c is a constant.

Now consider a witness point t of the δ-region arrangement corresponding to S. The following lemma states bounds for the witness points of the arrangement.

Lemma 3.2. *For all ε, ν, $0 < \varepsilon, \nu < 1$, and a sample set S of size $N \geq \frac{2d}{\varepsilon\nu} + d$, for a witness point $t \in \mathbb{R}^d$ of the arrangement of the δ-regions corresponding to the samples in S, the following holds:*

- $p_\delta(t) \leq \nu \Rightarrow \Pr(\tilde{p}_\delta(t) > (1+\varepsilon)\nu) \leq e^{-\frac{\varepsilon^2(N-d)\nu}{12}}$
- $p_\delta(t) \geq \nu \Rightarrow \Pr(|\tilde{p}_\delta(t) - p_\delta(t)| > \varepsilon p_\delta(t)) \leq 2e^{-\frac{\varepsilon^2\nu(N-d)}{16}}$.

Proof. Observe that Lemma 3.1 cannot be applied to the witness points directly since they depend on the experiment, i.e., the chosen samples. However, since a witness point depends only on $k \leq d$ samples, the remaining $\geq N - d$ samples are "random" for that point and we can apply Lemma 3.1 replacing N by $N - d$. More specifically:

Let $S_1, \ldots, S_k \in S$, $1 \leq k \leq d$, be the sample pairs whose δ-regions induce the witness point t. Consider the sample set $Q = S \setminus \{S_1, \ldots, S_k\}$, $|Q| = N - k$. The point t and the sample set Q are independent. Let $Z_Q(t)$ and $Z_S(t) = Z(t)$ denote the number of the δ-regions that cover t in the sample sets Q and S respectively, and $\tilde{p}_{\delta Q}(t) = Z_Q(t)/(N-k)$, $\tilde{p}_{\delta S}(t) = Z_S(t)/N = \tilde{p}_\delta(t)$. Since we consider closed neighborhoods, $Z_Q(t) = Z_S(t) - k$, $\tilde{p}_{\delta Q}(t) \leq \tilde{p}_{\delta S}(t)$ and

$$\tilde{p}_{\delta Q}(t) = \frac{Z_S(t) - k}{N - k} = \frac{Z_S(t)}{N}\frac{N}{N-k} - \frac{k}{N-k} \geq \tilde{p}_{\delta S}(t) - \frac{k}{N-k} \geq \tilde{p}_{\delta S}(t) - \frac{d}{N-d} \ .$$

Therefore,

$$|\tilde{p}_{\delta S}(t) - p_\delta(t)| \leq |\tilde{p}_{\delta Q}(t) - p_\delta(t)| + |\tilde{p}_{\delta S}(t) - \tilde{p}_{\delta Q}(t)| \leq |\tilde{p}_{\delta Q}(t) - p_\delta(t)| + \frac{d}{N-d} \ .$$

In case $p_\delta(t) \leq \nu$

$$\begin{aligned}
\Pr(\tilde{p}_{\delta S}(t) > (1+\varepsilon)\nu) &\leq P\left(\tilde{p}_{\delta Q}(t) + \frac{d}{N-d} > (1+\varepsilon)\nu\right) \\
&= P\left(\tilde{p}_{\delta Q}(t) > (1+\varepsilon)\nu - \frac{d}{N-d}\right) \\
&\leq P\left(\tilde{p}_{\delta Q}(t) > \left(1 + \frac{\varepsilon}{2}\right)\nu\right) \qquad \text{for } N \geq \tfrac{2d}{\varepsilon\nu} + d \\
&\leq e^{-\frac{(\varepsilon/2)^2(N-d)\nu}{3}} \qquad \text{by Lemma 3.1} \\
&= e^{-\frac{\varepsilon^2(N-d)\nu}{12}}
\end{aligned}$$

If $p_\delta(t) \geq \nu$:

$$\begin{aligned}
\Pr(|\tilde{p}_{\delta S}(t) - p_\delta(t)| > \varepsilon p_\delta(t)) &\leq P\left(|\tilde{p}_{\delta Q}(t) - p_\delta(t)| + \frac{d}{N-d} > \varepsilon p_\delta(t)\right) \\
&\leq P\left(|\tilde{p}_{\delta Q}(t) - p_\delta(t)| > \frac{\varepsilon}{2} p_\delta(t)\right) \qquad \text{for } N \geq \tfrac{2d}{\varepsilon\nu} + d \\
&\leq 2e^{-\frac{(\varepsilon/2)^2\nu(N-d)}{4}} \qquad \text{by Lemma 3.1} \\
&= 2e^{-\frac{\varepsilon^2\nu(N-d)}{16}}
\end{aligned}$$

□

In the above lemmas we used an additional parameter ν for the smallest value of $p_\delta(t)$ which we want to approximate well enough. If the considered transformation class allows to define an a priori lower bound for the maximum value of the hit probability function p_δ, then we can eliminate that parameter in the general error bound in the Theorem 3.1. For translations, rigid motions, and similarity transformations we show in Section 3.3 that there exists such lower bound, see Lemmas 3.4, 3.5 and 3.6.

Now we can prove Theorem 3.1:

Proof. (Of Theorem 3.1)

First, we show that for an arbitrary sequence of N random experiments performed by the algorithm, the probability that there exists a witness point t corresponding to this sequence, for which the estimate $\tilde{p}_\delta(t)$ of $p_\delta(t)$ is bad, can be bounded by a parameter η.

As we have seen above it is sufficient to consider k-subsets of the δ-regions with $1 \leq k \leq d$ in order to have at least one witness point in each cell of the arrangement. Any such tuple produces at most a constant number of witness points. We can enumerate all k-tuples of the δ-regions and, thus, all possible witness points corresponding to an arbitrary sequence of N experiments. For an arbitrary witness point t_{ij} (a witness point number j of the i-th δ-region tuple) we can apply Lemma 3.2 and, hence bound the probability that a sequence of N random experiments results in a bad approximation for t_{ij}. Note that if for some set of random experiments the i-th tuple does not have a witness point number j or does not have any witness points, the statement of the lemma trivially holds. In other words, according to Lemma 3.2 the probability that an arbitrary N-sequence gives a bad approximation for a witness point j produces by i-th tuple is at most $2e^{-\frac{\varepsilon^2\nu(N-d)}{16}}$.

Since there are at most N^d k-subsets of the δ-regions and, therefore, at most $c_1 N^d$ witness points, where c_1 is a constant, we have to apply the lemma at most $c_1 N^d$ times. Then, given some threshold value $0 < \nu < 1$, the probability that there exists a witness point t with $p_\delta(t) \geq \nu$ and $|\tilde{p}_\delta(t) - p_\delta(t)| > \varepsilon p_\delta(t)$ or with $p_\delta(t) < \nu$ and $\tilde{p}_\delta(t) > (1+\varepsilon)\nu$ is at most $c_1 N^d 2e^{-\frac{\varepsilon^2\nu(N-d)}{16}}$. A straightforward calculation shows that for $N \geq \frac{c_2}{\varepsilon^2\nu} \ln\left(\frac{1}{\varepsilon^2\nu}\right)$ with some suitable constant c_2 this value is at most $e^{-\frac{\varepsilon^2\nu(N-d)}{32}}$ (see Proposition 3.3 for a proof), which is less than η for $N \geq \frac{32}{\varepsilon^2\nu} \ln\frac{1}{\eta} + d$.

So the probability that there exists a witness point, for which the estimate of $p_\delta(t)$ is bad in the sense described above, is at most $\eta/2$ for

$$N \geq \frac{c}{\varepsilon^2 \nu} \ln\left(\max\left(\frac{1}{\eta}, \frac{1}{\varepsilon^2 \nu}\right)\right) \tag{3.1}$$

for some constant c. Observe that this is a combinatorial result which does not depend on the spatial position of witness points corresponding to a certain sequence of N experiments.

Since after N experiments the estimate $\tilde{p}_\delta$ is ε-accurate with probability at least $1-\eta$ for every witness point, it is in particular ε-accurate for the witness point t_{app}. Then for the case $p_\delta(t_{\text{opt}}) \leq \nu$ we have that $p_\delta(t_{\text{app}}) \leq p_\delta(t_{\text{opt}}) \leq \nu$. Therefore, the probability that $p_\delta(t_{\text{app}}) > (1+\varepsilon)\nu$ is at most η, which completes the proof of the part *(ii)* of the theorem.

If $p_\delta(t_{\text{opt}}) > \nu$ then, by Lemma 3.1, after N experiments the probability of a bad estimation of $p_\delta(t_{\text{opt}})$ is also at most η. Combining these error bounds we get

$$\begin{aligned} \tilde{p}_\delta(t_{\text{app}}) &\geq \tilde{p}_\delta(t_{\text{opt}}) && \text{since } t_{\text{app}} \text{ maximizes } \tilde{p}_\delta(t) \\ &\geq (1-\varepsilon) p_\delta(t_{\text{opt}}) && \text{with probability } \geq \eta \text{ by Lemma 3.1} \end{aligned}$$

and

$$\begin{aligned} \tilde{p}_\delta(t_{\text{app}}) &\leq (1+\varepsilon) p_\delta(t_{\text{app}}) && \text{with probability } \geq \eta \text{ by Lemma 3.2} \\ &\leq (1+\varepsilon) p_\delta(t_{\text{opt}}) && \text{since } t_{\text{opt}} \text{ maximizes } p_\delta(t) \end{aligned}$$

Therefore, $|\tilde{p}_\delta(t_{\text{app}}) - p_\delta(t_{\text{opt}})| \leq \varepsilon p_\delta(t_{\text{opt}})$ with probability at least $1-2\eta$ (part *(i)* of the theorem).

For the part *(iii)* of the theorem we show in Lemmas 3.4, 3.5 and 3.6 that in case of translations, rigid motions with Approach 1, and similarity transformations for any two shapes and a tolerance bound δ there always exists a transformation t, such that $p_\delta(t) \geq \frac{\delta^2}{m^2}$, where $m = \max(L_A, L_B, n\delta)$. Then for $\nu = \frac{\delta^2}{m^2}$ the maximum of $p_\delta(t)$ is $p_\delta(t_{\text{opt}}) \geq \nu$.

Plugging $\frac{\delta^2}{m^2}$ for ν in Formula (3.1) we obtain that after

$$N \geq c\frac{m^2}{\varepsilon^2 \delta^2} \ln\left(\max\left(\frac{1}{\eta}, \frac{m^2}{\varepsilon^2 \delta^2}\right)\right)$$

experiments it holds with probability at least $1-2\eta$ that $|\tilde{p}_\delta(t_{\text{app}}) - p_\delta(t_{\text{opt}})| \leq \varepsilon p_\delta(t_{\text{opt}})$. □

It remains to prove the following proposition for completeness of the proof:

Proposition 3.3. *Let c, k, d be positive real numbers and let $m = \max\left\{\frac{2d+1}{k}, c\right\}$. For all $N \geq m \log m$ holds*

$$cN^d \leq e^{kN} \quad .$$

Proof. Let m be a positive real number and $N_0 = m \log m$. The claim of the proposition for N_0 is equivalent to $\frac{e^{kN_0/d}}{N_0} \geq c^{1/d}$. Evolving the left side of this inequality we get

$$\begin{aligned} \frac{e^{kN_0/d}}{N_0} = \frac{e^{km \log m/d}}{m \log m} = \frac{m^{mk/d}}{m \log m} &\geq m^{\frac{mk}{d}-2} \\ &\geq m^{1/d} \qquad \text{if } m \geq \frac{2d+1}{k} \\ &\geq c^{1/d} \qquad \text{if } m \geq c \end{aligned}$$

which implies the claim for N_0 and thus for all $N \geq N_0$. □

Running time

The running time of the algorithm consists of the time needed to generate N random samples denoted by $T_{\text{gen}}(n, N)$, where n is the number of curves in the shape, and the time needed to determine the depth of the arrangement of N δ-regions denoted by $T_{\text{depth}}(N)$.

As mentioned in Section 2.1 we assume that every smooth curve piece in our curve sets has a constant speed parameterization, which is either given or can be reconstructed in constant time. The latter is the case for line segments, and thus, for polygonal curves. Under this assumption a random point on a curve can be generated in constant time. For generating a random point from a set of n curves we first select a curve randomly with probability proportional to the relative length of the curve and then take a random point from the selected curve uniformly with respect to length, which is equivalent to selecting a random value uniformly from a real numbers interval.

The selection of a random segment can be trivially done in time linear in n. If we first compute the relative curve lengths and record the corresponding probabilities to allow for binary search during the generation process we get preprocessing time linear in n and $O(\log n)$ generation time for a single point. Therefore, $T_{\text{gen}}(n, N) = O(n + N \log n)$.

In order to determine the depth of the arrangement of N δ-regions we can construct this arrangement and during the construction keep record of the depth of the cells. Then at the end of the construction algorithm we know the depth of the deepest cell. For general metrics L_p and the considered classes of transformations the boundaries of δ-regions are defined by semi-algebraic varieties. Every sample pair induces a constant number of such varieties. And for all transformation classes, except for rigid motions, we consider the arrangement of the varieties produced by N experiments in the corresponding d-dimensional transformation space. By Basu et al. [11], the corresponding arrangement can be constructed and traversed in time $O(N^{d+1}p^{O(d^2)})$, where p denotes the degree of polynomials describing the boundaries of the δ-regions and depends on the chosen distance metric L_p. Since once the dis-

tance metric is chosen, the degree of polynomials can be considered as constant, we can simplify the expression for the running time to $T_{\text{depth}}(N) = O(N^{d+1})$.

In case of rigid motions the boundaries of δ-regions are described by polynomials in four variables, i.e., we consider semi-algebraic hypersurfaces in four (that is in $d+1$) dimensions. But we are interested in the deepest cell of the arrangement induced on a three (d) dimensional variety. With the algorithm by Basu et al. [12] it can be done in time $O(N^{d+1}p^{O(d+1)})$, where p is the degree of polynomials, which depends on the chosen distance metric L_p. Considering p as a constant we get the same time bounds for rigid motions as for translations and similarity maps, $T_{\text{depth}}(N) = O(N^{d+1})$.

Alternatively, we can compute the intersection of the boundaries for all k-subsets of δ-regions, $1 \leq k \leq d$, and choose one point in each connected component of the intersection. Since we assumed that the distance metric is an algebraic function of constant degree, the intersection computation means solving a system of polynomial equations with constant number of equations and polynomials of constant degree, which can be performed numerically in constant time. Then for every computed witness point we can count the number of δ-regions containing it. Thus, we have $O(N^d)$ intersection computations and for each of the $O(N^d)$ witness points $O(N)$ time cost for counting the number of δ-regions covering it.

Summarizing these results and using Theorem 3.1 we obtain the following running time of the algorithm:

Theorem 3.2. *Given two shapes A and B modeled by finite sets of n rectifiable curves in the plane of total lengths L_A and L_B respectively, parameters ε, η, ν, $0 < \varepsilon, \eta, \nu < 1$, and a transformation class $\mathcal{T}$ of dimension d, such that the intersection of the boundaries of any up to d δ-regions in $\mathcal{T}$ has constant complexity. Let t_{opt} denote the transformation maximizing $p_\delta(t)$. In time $O(n + N \log n + N^{d+1})$ the generic probabilistic algorithm computes a transformation t_{app} such that $|\tilde{p}_\delta(t_{\text{app}}) - p_\delta(t_{\text{opt}})| \leq \varepsilon p_\delta(t_{\text{opt}})$ or determines that no transformation t with $p_\delta(t) > \nu$ exists with error probability at most η. The required number of experiments is $N \in O\left(\frac{1}{\varepsilon^2 \nu} \ln\left(\max\left(\frac{1}{\eta}, \frac{1}{\varepsilon^2 \nu}\right)\right)\right)$.*

For translations, rigid motions (Approach 1), and similarity transformations the algorithm finds an ε-approximation of $p_\delta(t_{\text{opt}})$ with probability at least $1 - \eta$, the required number of experiments is $N \in O\left(\frac{m^2}{\varepsilon^2 \delta^2} \ln\left(\max\left(\frac{1}{\eta}, \frac{m^2}{\varepsilon^2 \delta^2}\right)\right)\right)$, where $m = \max(L_A, L_B, n\delta)$.

Observe that, at least for sufficiently small values of δ, the runtime of the algorithm depends much more on the parameters ε and η than on the combinatorial input size n, which is only needed in the preprocessing and the drawing of random samples.

The running time of the algorithm is actually better than that stated in Theorem 3.2 for translations and for similarities in combination with the L_1 and L_∞ metrics. In case of translations, the δ-regions are pseudo-disks and their arrangement can be constructed straightforwardly in time $O(N^2)$. For similarities in combination

with the L_1 or L_∞ metric the δ-regions in transformation space are bounded by a constant number of 3-dimensional hyperplanes. Using the algorithm of Edelsbrunner et al. [23] the arrangement of N such δ-regions can be constructed in $O(N^4)$ time. These observations are summarized in the following remarks:

Remark 3.2. *Under the conditions of Theorem 3.2 for the case of translations the generic probabilistic algorithm computes a good approximation of the maximum of $p_\delta(t)$ in time $O(n + N \log n + N^2)$, where a good approximation is to be understood as stated in the Theorem.*

Remark 3.3. *Under the conditions of Theorem 3.2 for the case of similarity transformations in combination with L_1 or L_∞ metric in image space the generic probabilistic algorithm computes a good approximation of the maximum of $p_\delta(t)$ in time $O(n + N \log n + N^4)$, where a good approximation is to be understood as stated in the Theorem.*

In order to achieve a speed up of the algorithm a combination with the depth approximation algorithm by Aronov and Har-Peled [9] is possible. Given a set of N object in $\mathbb{R}^d$ whose arrangement has depth $\mathcal{D}$ and a prespecified parameter $\varepsilon > 0$, their algorithm finds a point of depth at least $(1-\varepsilon)\mathcal{D}$ in $O\left(N + T_{\mathrm{DT}}(N, \varepsilon^{-2} \log n) \log n\right)$ expected time, where $T_{\mathrm{DT}}(N, k)$ is the running time of a depth thresholding algorithm. A depth thresholding algorithm takes a set $\mathcal{S}$ of N objects and an integer $k > 0$ and returns the depth of the arrangement of $\mathcal{S}$ together with a witness point if this depth is at most k, or tells that the depth is greater than k.

It is easy to verify that in combination with depth approximation the probabilistic algorithm gives an approximation of $p_\delta(t_{\mathrm{opt}})$ with an error at most 2ε: Let t^* be the transformation determined by the depth approximation algorithm, t_{app} a transformation maximizing $\tilde{p}_\delta(t)$, and t_{opt} a transformation maximizing $p_\delta(t)$, then

$$\begin{aligned}
\tilde{p}_\delta(t^*) &\geq (1-\varepsilon)\tilde{p}_\delta(t_{\mathrm{app}}) \quad \text{by the property of depth approximmation algorithm} \\
&\geq (1-\varepsilon)^2 p_\delta(t_{\mathrm{opt}}) \quad \text{by Theorem 3.1} \\
&\geq (1-2\varepsilon) p_\delta(t_{\mathrm{opt}}) \quad \text{for } 0 < \varepsilon < 1 \ ,
\end{aligned}$$

and

$$\tilde{p}_\delta(t^*) \leq \tilde{p}_\delta(t_{\mathrm{app}}) \leq (1+\varepsilon) p_\delta(t_{\mathrm{opt}}) \ .$$

Thus, provided a fast depth thresholding algorithm exists, this combination would result in a significant speed up of the probabilistic algorithm. For translations, that is for an arrangement of pseudo-disks in the plane, a fast thresholding algorithm is known to exist [9], which results in running time $T_{\mathrm{depth}}(N) = O(N\varepsilon^{-2} \log N)$.

3.2.4 Generic Description

The generic probabilistic algorithm as described in Section 3.1 is not restricted to sub-classes of the affine transformations but can be applied to matching curve sets

with respect to a broader class of transformations, e.g., projective or even non-linear transformations. In this section we summarize which steps of the algorithm need to be specified for a transformation class $\mathcal{T}$ with d degrees of freedom, and the conditions that need to be fulfilled for the approximation error guarantees in Theorem 3.1 and the running time bound in Theorem 3.2 to hold.

First, we need to specify the samples generated during one random experiment and the corresponding δ-region. For transformation classes of dimension $d \geq 2$, i.e., with two ore more degrees of freedom, one can always use a single randomly selected point of each shape as the random sample, i.e., $S_A = a \in A$ and $S_B = b \in B$. Then, in general, the set of transformations that map S_B to S_A is a $(d-2)$-dimensional variety in the d dimensional transformation space. Larger samples with up to d parameters constrain further degrees of freedom and so reduce the dimension of the corresponding set of transformations. The larger samples also induce δ-regions of smaller volume.

From the general analysis of the objective function of the algorithm it follows: If the random sample consists only of points of the shapes, in particular of a single point, then the algorithm maximizes the measure of the set of point pairs that are δ-close to each other after the chosen transformation is applied. For the samples containing different characteristics assigned to the points the matching criterion depends on these parameters and the definition of the δ-regions. The additional parameters can help to reduce the search space for the transformation resulting in a good match, but, as for example in the case of rigid motions (see Section 3.3.2), might lead to matching results that are different from those based only on the proximity of points.

The δ-regions induced by a random sample have to be specified by a constant number of polynomial equations and inequalities. Then the algorithm by Basu et al. [11] can be used to compute the arrangement of δ-regions and determine the deepest cell of the arrangement. For some transformation classes the intuitive parameterization of the transformation space results in a non-algebraic description of δ-regions, e.g., for rigid motions. In such cases one should examine the possibility of reparameterization and embedding in a higher dimensional space to obtain a semi-algebraic description of the δ-regions. Then the algorithm by Basu et al. [12] for computing the arrangement of semi-algebraic sets on a variety gives the desired result. In both cases the computation time of the deepest cell of the arrangement of δ-regions is $O(N^{d+1})$, where N is the number of random experiments. If the δ-region description contains only linear equations, that is, a δ-region is bounded by a set of hyperplanes, the arrangement can be constructed an traversed in time $O(N^d)$ with the algorithm by Edelsbrunner et al. [23].

If the definition of the δ-regions with polynomial equations or inequalities is not possible, one has to specify the way to find a transformation covered by the most δ-regions corresponding to the sample set. In this case the computation time of the deepest cell of the arrangement might vary from that stated in the Theorem 3.2. Although, if the definition of a δ-region fulfills the condition that the intersection of

the boundaries of any k δ-regions has a constant number of connected components, and given the possibility to compute a point in each of the connected components, the statement of the Theorem 3.2 still holds: By considering all k, $1 \leq k \leq d$, tuples of δ-regions and taking one point in each of the connected components in the intersection of their boundaries, we get a set W of witness points of the arrangement (defined in Section 3.2.3). The set W contains $O(N^d)$ points, and for every cell of the arrangement there exists at least one point in W that lies on the boundary of that cell. Then for each point in W we can count the number of δ-regions covering that point. The computation of the deepest cell of the arrangement has then the time complexity of $O(N^d T_{\text{con}} + N^{d+1} T_{\text{cov}})$, where T_{con} denotes the time needed to compute a point in each connected component of the boundary of k δ-regions, and T_{cov} the time to test whether a point lies inside a δ-region. If the last two operations can be performed in constant time, we get the desired computation bounds.

Finally, for the approximation error bound that is independent of a threshold value one needs to show that for the given class of transformations and for any two shapes A and B there exists some value ν, which depends on the distance tolerance value δ and properties of the shapes, such that the maximum of the underlying probability distribution $p_\delta(t)$ is at least the value ν. That is, statements equivalent to those of Lemmas 3.4, 3.5 and 3.6 need to hold for the considered class of transformations. If the value ν is proportional to $\frac{\delta^2}{m^2}$, where δ is the distance tolerance value and $m = \max(L_A, L_B, n\delta)$ with L_A, L_B being the total length of the shapes A and B respectively, and n the number of curve segments comprising the shapes, then the bound on the number of experiments is the same as for translations, rigid motions and similarity transformations given by Theorem 3.1.

Summarizing, we can say that in order to apply the probabilistic matching method to matching shapes modeled as sets of curves under an arbitrary transformation class with d degrees of freedom one need to complete the following steps:

- Specify the random sample taken from each shape within one random experiment. A sample can consist of up to d parameters.
- Provide a description of the corresponding δ-region as a semi-algebraic variety. If such description is not possible, provide a method for finding the deepest cell in the arrangement of δ-regions.
- For the threshold independent relative error bound show that for any two shapes there exists an a priori lower bound for the maximum value of the hit probability function $p_\delta(t)$.
- Adapt the analysis of the running time of the algorithm if the sample generation procedure or the computation of the deepest cell of the arrangement differ from that described in Section 3.2.3.

We complete the description of the algorithm according to the described steps for

translations, rigid motions, similarity, homothety, shear and affine transformations in Section 3.3.

3.3 Details for Subclasses of Affine Transformations

The transformation classes considered in this section are subclasses of affine transformations. An affine transformation t is defined by a linear transformation matrix

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \tag{3.2}$$

and a translation vector $v = (v_x, v_y)$, and maps a point $b \in \mathbb{R}^2$ to a point $t(b) = Mb+v$. Then the condition $t(b) \in U_\delta(a)$ for two points a and b and a transformation t means $\text{dist}(t(b), a) \leq \delta$. For general L_p metrics, since the distance function is always positive, the condition $\text{dist}(t(b), a) \leq \delta$ is equivalent to $\text{dist}^p(t(b), a) \leq \delta^p$. Then, for even values of p the distance condition can be described by one inequality of degree p:

$$(m_1 b_x + m_2 b_y + v_x - a_x)^p + (m_1 b_y + m_2 b_x + v_y - a_y)^p \leq \delta^p \tag{3.3}$$

For odd values of p the distance condition induces four inequalities of degree p:

$$\begin{aligned} (m_1 b_x + m_2 b_y + v_x - a_x)^p + (m_3 b_y + m_4 b_x + v_y - a_y)^p &\leq \delta^p \\ -(m_1 b_x + m_2 b_y + v_x - a_x)^p + (m_3 b_y + m_4 b_x + v_y - a_y)^p &\leq \delta^p \\ (m_1 b_x + m_2 b_y + v_x - a_x)^p - (m_3 b_y + m_4 b_x + v_y - a_y)^p &\leq \delta^p \\ -(m_1 b_x + m_2 b_y + v_x - a_x)^p - (m_3 b_y + m_4 b_x + v_y - a_y)^p &\leq \delta^p \end{aligned} \tag{3.4}$$

And for the L_∞ metric we get four linear inequalities:

$$\begin{aligned} m_1 b_x + m_2 b_y + v_x - a_x &\leq \delta \\ -(m_1 b_x + m_2 b_y + v_x - a_x) &\leq \delta \\ m_3 b_y + m_4 b_x + v_y - a_y &\leq \delta \\ -(m_3 b_y + m_4 b_x + v_y - a_y) &\leq \delta \end{aligned} \tag{3.5}$$

These distance constraints contribute to the description of the δ-regions in the following analysis of the effect of the matching algorithm for different transformation classes.

3.3.1 Translations

The first class of transformations that we consider are translations. The translation space $\mathcal{T}$ is two-dimensional. Two points $a, b \in \mathbb{R}^2$ define uniquely a translation t that maps b to a, $t = a - b$. Therefore, a random sample of each shape in step one of our algorithm consist of a single point and the sample space is $\Omega = A \times B$. The set

$M_\delta(t)$ is then the set of the point pairs of the two shapes that are mapped to the δ-neighborhood of each other by the transformation t, formally,

$$M_\delta(t) = \{(a,b) \in A \times B \mid t(b) \in U_\delta(a)\} \quad . \tag{3.6}$$

With an L_p metric as the underlying distance function for the points in the plane a δ-region corresponding to a sample pair a, b is a convex region defined by inequalities (3.3) – (3.5) depending on p, where the matrix M defined in (3.2) is the identity matrix with $m_1 = m_4 = 1$ and $m_2 = m_3 = 0$.

To maximize the measure of the set $M_\delta(t)$ means to find a transformation that maps largest possible parts of the shapes into proximity of each other, which comes close to the intuitive notion of matching of two shapes.

Observe that there is a direct connection between the set $M_\delta(t)$ and a free space diagram, which was first defined for polygonal curves by Alt and Godau in [2]. Let $f : I \to \mathbb{R}^2,\ g : J \to \mathbb{R}^2$ be two curves, with parameter intervals $I, J \subset \mathbb{R}$. The set $F_\delta(f,g) := \{(s,r) \in I \times J \mid \mathrm{dist}(f(s) - g(r)) \leq \delta\}$ denotes the *free space* of f and g, where $\mathrm{dist}(\cdot,\cdot)$ denotes the distance measure in the image space. The partition of $I \times J$ into regions belonging or not belonging to $F_\delta(f,g)$ is called the *free space diagram*.

A set of curves can be parameterized over the interval $[0,1] \subset \mathbb{R}$ in the following way: Let the set A be composed of k curve segments of lengths $l_1, \ldots, l_k$ and let L denote the total length of A. Subdivide the interval $[0,1]$ into k subintervals $I_j = [r_{j-1}, r_j]$ for $1 \leq j \leq k$ of size proportional to the lengths of curves with $r_0 = 0$ and $r_j = \sum_{i=1}^{j} l_i/L$ for $1 \leq j \leq k$. Since we assumed that the curves are given by natural parameterization $\mathcal{C}_j : [0, l_j] \to \mathbb{R}^2$, the constant speed reparameterization over the intervals I_j can be easily constructed as $\mathcal{C}'_j : [r_{j-1}, r_j] \to \mathbb{R}^2$ with $\mathcal{C}'_j(r) = \mathcal{C}_j\left((r - r_{j-1}) \cdot L\right)$. Then the whole set A is a piecewise function from $[0,1]$ to $\mathbb{R}^2$ and equals $\mathcal{C}'_j$ on a subinterval I_j. The order of curves can be arbitrary, since we consider the shapes as the points in the plane without orientation or ordering.

Thus, the shapes A and B can be regarded as functions $A : [0,1] \to \mathbb{R}^2,\ B : [0,1] \to \mathbb{R}^2$. Then, the free space of the set A and the set B transformed by t is defined as $F_\delta(A, t(B)) := \left\{(s,r) \in [0,1]^2 \mid \mathrm{dist}(A(s) - t(B(r))) \leq \delta\right\}$, see Figure 3.4 for an example.

It is easy to see, that there is a one-to-one correspondence between the set $M_\delta(t)$ defined above and the set $F_\delta(A, t(B))$, namely a parameter pair (s,r) is in $F_\delta(A, t(B))$ if and only if the corresponding pair of points $(A(s), t(B(r)))$ is in $M_\delta(t)$. The measure of the set $F_\delta(A, t(B))$ is exactly the measure of $M_\delta(t)$ normalized by the total measure of sampling space, that is, $p_\delta(t) = \frac{|M_\delta(t)|}{|A \times B|} = |F_\delta(A, t(B))|$.

Another observation that we make is, if a translation t corresponds to the sample pair $s = (a,b)$ then a transformation t' is covered by the δ-region corresponding to s exactly if the distance between vectors t and t' is at most δ under the same distance measure as for points. That means every sample pair produces a δ-region of the same size and shape in the space of translations. Thus, in case of translations we do not need to record a sample pair (a,b), but only the translation $t = a - b$ and

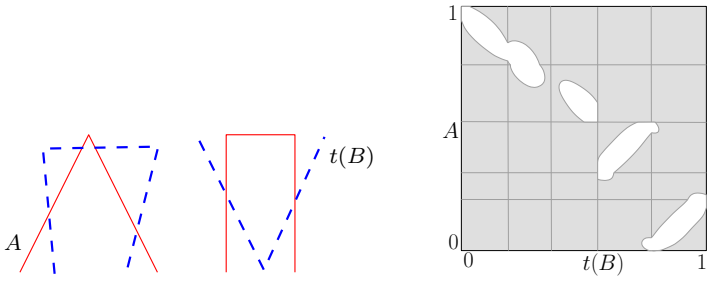


Figure 3.4: Free space diagram of two sets of curves. White regions denote the free space $F_\delta(A, t(B))$.

then consider the δ-neighborhoods of the translation vectors, which are defined in the same way as the δ-neighborhoods of points.

For a general L_p-metric, $1 \leq p \leq \infty$, the arrangement of the δ-regions is an arrangement of pseudo-disks in the plane, that is, the boundaries of any two objects have at most two intersection points. The arrangement can be computed in $O(N^2)$ time, where N is the number of objects.

If for each sample pair (a, b) we record a translation that maps b to a instead of the corresponding δ-region, we get a certain distribution of "votes" in translation space. We show in Section 4.1 that the density function of this probability distribution is exactly the weighted generalized Radon transform of the shape A with respect to shape B, [29, 30, 57]. The hit probability induced by δ-regions corresponds to a "smoothed" version of the generalized Radon transform. In fact, the hit probability function is a convolution of the normalized generalized Radon transform with a function that is constant over a δ-neighborhood of the origin and zero elsewhere and integrates to one.

Finally, we complete the proof of the part *(iii)* of Theorem 3.1 for translations. In the next Lemma it is shown that for any two shapes and a parameter value δ there exists an a priori lower bound ν such that the maximum of the hit probability function p_δ is greater or equal ν.

Lemma 3.4. *Given two shapes A and B modeled by finite sets of n rectifiable curves with total lengths L_A, L_B respectively and a tolerance value $\delta > 0$. If the class of the allowed transformations $\mathcal{T}$ is the group of translations then $\max_{t\in\mathcal{T}} p_\delta(t) \geq \nu$, where $\nu = \frac{\delta^2}{m^2}$ and $m = \max(L_A, L_B, n\delta)$.*

Proof. We show that for any two shapes A and B there exists a translation vector t_x such that $p_\delta(t_x) \geq \frac{\delta^2}{m^2}$. Then the maximum $\max_{t\in\mathcal{T}} p_\delta(t) \geq p_\delta(t_x) \geq \frac{\delta^2}{m^2}$.

If the shape A contains a curve segment of length at least δ, let s_a denote a subsegment of that segment with length exactly δ. Otherwise let s_a denote the longest curve segment in A. s_a has length at least $\frac{L_A}{n} \leq \delta$. Similarly s_b denotes a subsegment of length δ or the longest segment of B with length at least $\frac{L_B}{n} \leq \delta$. Let x_a, x_b denote the centers of the segments s_a, s_b, respectively, and t_x the translation

vector $x_a - x_b$. For an arbitrary point p_a of s_a and an arbitrary point p_b of s_b the distances between p_a and x_a and between p_b and x_b are at most $\delta/2$. Then it holds that

$$\begin{aligned}\operatorname{dist}(p_a, p_b + t_x) &\leq \operatorname{dist}(p_a, x_a) + \operatorname{dist}(x_a, p_b + t_x) \\ &\leq \delta/2 + \operatorname{dist}(x_a, p_b + t_x) \\ &= \delta/2 + \operatorname{dist}(x_b + t_x, p_b + t_x) \\ &= \delta/2 + \operatorname{dist}(x_b, p_b) \\ &\leq \delta \ ,\end{aligned}$$

therefore t_x is covered by the δ-region corresponding to p_a, p_b. Thus, $s_a \times s_b$ is a subset of $M_\delta(t_x)$ and $|M_\delta(t_x)| \geq \min\left(\delta^2, \frac{L\delta}{n}, \frac{L_A L_B}{n^2}\right)$, where $L = \min(L_A, L_B)$. Then $p_\delta(t_x) \geq \min\left(\frac{\delta^2}{L_A L_B}, \frac{L\delta^2}{(n\delta) L_A L_B}, \frac{L_A L_B \delta^2}{(n\delta)^2 L_A L_B}\right) \geq \frac{\delta^2}{m^2}$. □

3.3.2 Rigid Motions

The space of rigid motions $\mathcal{T}$ is three dimensional, $\mathcal{T} = [0, 2\pi) \times \mathbb{R}^2$. For two points in the plane there is no unique rigid motion that maps one point to the other, rather for every rotation angle we can find a unique translation vector such that the resulting rigid motion performs the desired mapping. For two points the set of rigid motions that map one of the points to the other is, therefore, a one dimensional curve in three dimensional transformation space.

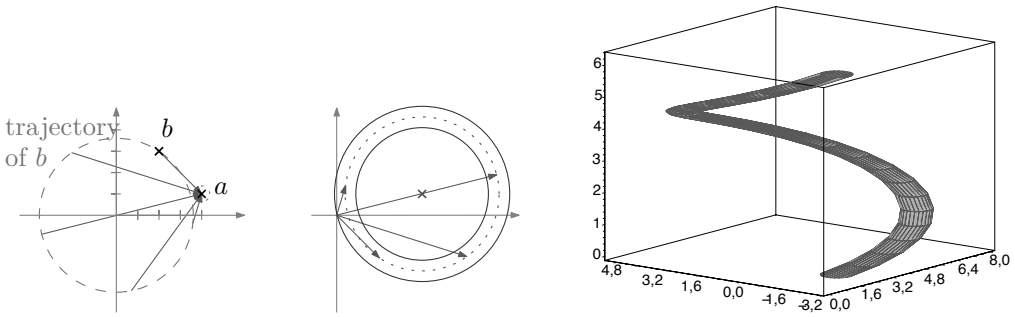
As mentioned in Section 3.2.3, for computational reasons we will also view the space of rigid motions as a three dimensional subspace of a four dimensional space of similarity transformations. Recall, that the space of similarity transformations is parameterized by (m_1, m_2, v_x, v_y), where (v_x, v_y) is the translation vector, and m_1, m_2 describe the rotation and scaling matrix $M = \begin{pmatrix} m_1 & m_2 \\ -m_2 & m_1 \end{pmatrix}$, with $m_1 = k\cos\alpha$, $m_2 = -k\sin\alpha$. Since rigid motions are similarity transformations with scaling factor $k = 1$, the subspace of rigid motions is an algebraic variety described by the polynomial

$$m_1^2 + m_2^2 - 1 = 0 \tag{3.7}$$

In the following the intuitive view as a three dimensional space with parameterization by angle and translation vector is used to explain the functioning of the algorithm and the effect of the different approaches on matching results.

Approach 1: In Approach 1 for the rigid motions as described in section 3.1 we use a single random point of each shape $a \in A$ and $b \in B$ as a sample in one random experiment and record a δ-region in the space of rigid motions as a set of transformations that map the point b into the δ-neighborhood of the point a. The sample space is in this case $\Omega = A \times B$. A δ-region corresponding to a sample

$(a, b) \in \Omega$ has a shape of a spiral tube, where the projection to the translation vector plane has a shape of an annulus with center a and radius $\|\bar{b}\|$ of width δ as shown in Figure 3.5.



(a) (b) (c)

Figure 3.5: δ-region in the space of rigid motions corresponding to a pair of points $a \in A$ and $b \in B$. (a) Points a and b with translation vectors corresponding to some rotated positions of b. (b) Projection of the δ-region to the translation plane for $\delta = 1$. (c) δ-region in the 3-dimensional space of rigid motions.

With respect to the four dimensional parametrization by (m_1, m_2, v_x, v_y) a δ-region is a semi-algebraic set described, depending on the chosen distance metric L_p, either by one inequality (3.3) of degree p for even values of p, or by four inequalities (3.4) of degree p for odd values of p, or by four linear inequalities (3.5) in case of L_∞, which are constrained to a three dimensional variety defined by Equation (3.7).

Now by taking a rigid motion covered by the most neighborhoods we find a transformation that maximizes the measure of the set of point pairs that are mapped into the δ-neighborhood of each other by this transformation, just as in the case of translations. That is, the set of sample pairs that "vote" for a rigid motion t is

$$M_\delta(t) = \{(a, b) \in A \times B \mid t(b) \in U_\delta(a)\}$$

and the probability of t to be covered by a region corresponding to a randomly selected sample pair is $p_\delta(t) = \frac{|M_\delta(t)|}{|A \times B|} = |F_\delta(A, t(B))|$.

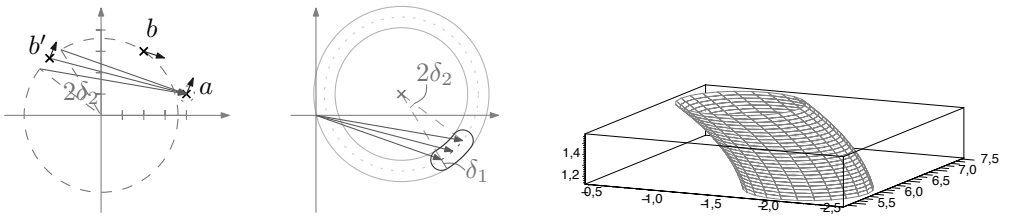
Approach 2: In Approach 2 a sample of a shape taken within one random experiment in the first step of the generic algorithm consists of a random point of the shape and the angle defined by the (interpolated) direction of the tangent line at that point. The sample space Ω is a subset of $A \times B \times [0, 2\pi)^2$.

A motivation for this approach is that if we would, somehow, have a unit length direction vector associated with each point, then the rotation angle that maps one direction vector to the other would be uniquely defined and so the rigid motion. And a natural direction associated with a point on a curve is the direction of the tangent line to the curve at that point.

Note that, since in general we consider the shapes as points in the plane without explicit direction information, the orientation of the tangent line depends on the chosen parameterisation, which is not a part of the visible shape information and, therefore, is not important for the similarity evaluation. For this reason, each point pair a, b actually generates two random sample pairs: $S_A = (a, \theta_a),\ S_B = (b, \theta_b)$, where θ_a and θ_b denote the angle of inclination of the normal of the lines tangent to A and B at the points a and b, respectively; and $S_A = (a, \theta_a), S'_B = (b, \theta'_b)$, where θ_a is defined as above, and $\theta'_b = \theta_b + \pi$. So in one random experiment we produce two random samples and the corresponding δ-regions. In this way we record the transformations that map the tangent line at the point b to a line parallel to, or to the line that forms an angle at most δ_2 with, the tangent line at the point a independent of the line orientations.

We defined a δ-region in the transformation space corresponding to a sample pair as the set of transformations that map each component of one sample into a δ-neighborhood of the corresponding component of the other sample. So far we interpreted δ as a distance parameter for points but it does not make much sense for the directions. Therefore, in this case we need two parameters $\delta = (\delta_1, \delta_2)$, where δ_1 controls the distance that the points are allowed to be apart and still be considered close, and δ_2 represents the maximum difference in the directions that are still considered to be similar.

A pair of samples $S_A = (a, \theta_a),\ S_B = (b, \theta_b)$ gives us the following neighborhood in the space of rigid motions: the rotation angles are restricted to the interval $I = [\alpha - \delta_2, \alpha + \delta_2]$, where $\alpha = \theta_a - \theta_b$, and for each $\alpha' \in I$ the allowed translations are v' such that $\text{dist}(\alpha'(b) + v', a) \leq \delta_1$, see Figure 3.6 for an illustration.



(a) (b) (c)

Figure 3.6: δ-region in the space of rigid motions corresponding to a pair of points $a \in A$ and $b \in B$. (a) Points a and b with translation vectors corresponding to some rotated positions of b. (b) Projection of the δ-region to the translation plane for $\delta_1 = 1$. (c) δ-region in the 3-dimensional space of rigid motions.

In the four dimensional parameterization the restriction of the rotation angle α'

to the interval I implies the following four inequalities:

$$m_1 \leq \max_{\alpha' \in I} \cos \alpha'$$
$$m_1 \geq \min_{\alpha' \in I} \cos \alpha'$$
$$m_2 \leq \max_{\alpha' \in I} \sin \alpha'$$
$$m_2 \geq \min_{\alpha' \in I} \sin \alpha'$$

Note that we do not compute the angles θ_a, θ_b and α explicitly. Instead, we assume that the tangent lines at the points a and b are given in Hesse standard form, i.e., the tangent line at point a is given as $g_a : u_a x + v_a y + w_a = 0$, and the tangent line at point b is given as $g_b : u_b x + v_b y + w_b = 0$, where $u_a = \cos \theta_a$, $v_a = \sin \theta_a$ and $u_b = \cos \theta_b$, $v_b = \sin \theta_b$. Then, by simple trigonometric rules, the cosine and sine of the angle α are computed as: $\cos \alpha = u_a u_b + v_a v_b$ and $\sin \alpha = v_a u_b - u_a v_b$. Similarly, the cosine and sine of the endpoints of the interval I are $\cos(\alpha \pm \delta_2) = \cos \alpha \cos \delta_2 \mp \sin \alpha \sin \delta_2$ and $\sin(\alpha \pm \delta_2) = \sin \alpha \cos \delta_2 \pm \cos \alpha \sin \delta_2$. The tolerance angle δ_2 is assumed to be specified by its cosine value, and the value of its sine can be computed as $\sin \delta_2 = \sqrt{1 - \cos^2 \delta_2}$.

Finally, it can be easily checked whether the interval I contains $0, \pi/2, \pi$ or $3\pi/2$ by testing the signs of the sine and cosine values of the endpoints of the interval. If $k\pi/2 \in I$ for $k \in \{0, 1, 2, 3\}$ the corresponding maximum or minimum value is 1 or -1, otherwise minimum and maximum values in the inequalities above are attained at the endpoints of the interval.

The distance condition is described by inequalities (3.3) – (3.5), depending on the underlying distance metric L_p, with $m_3 = -m_2$ and $m_4 = m_1$ and δ_1 as the tolerance bound. Thus, a δ-region is a semi-algebraic set defined by five or eight polynomials, which is constraint to the algebraic variety of dimension three, described by a polynomial of degree two, Equation (3.7).

By taking the transformation that is covered by the most neighborhoods we maximize the measure of the set of point pairs, that are close to each other (have distance at most δ_1) and have similar tangent directions (the directions differ by at most δ_2). Note that this definition of a good match is different from that we used for the translations, since for translations we did not take the direction of curves into account.

However, there is still a connection to the free space diagram of the shapes. Consider again shapes to be parameterized over the interval $[0, 1]$, $A : [0, 1] \to \mathbb{R}^2$, $B : [0, 1] \to \mathbb{R}^2$. Let function $g : [0, 1]^2 \to \mathbb{R}$ denote the distance between the corresponding points on the shapes with respect to the chosen metric, $g(s, r) = \text{dist}(A(s), B(r))$. The free space $F_{\delta_1}(A, B)$ in the free space diagram is defined as the set of pairs (s, r) for which $g(s, r) \leq \delta_1$. The additional condition about the closeness of tangent directions can be expressed by a function $h : [0, 1]^2 \to \mathbb{R}$ which assigns to two parameter values the absolute difference of the tangent directions of the corresponding points. Let $\theta_A(s), \theta_B(r)$ denote the angles of slope of the tangent

lines to the points $A(s)$ and $B(r)$ respectively. Then the function h can be written as $h(s,r) = |\theta_A(s) - \theta_B(r)|$. Similar to the free space F_{δ_1} with respect to distance function g we can define free space H_{δ_2} with respect to direction distance function h, $H_{\delta_2}(A,B) = \left\{(s,r) \in [0,1]^2 \mid h(s,r) \leq \delta_2\right\}$. The set of the "good" samples for a fixed rigid motion t then corresponds to the intersection of the two free spaces (see Figure 3.7 for an illustration):

$$M_{\delta_1,\delta_2}(t) = \{(A(s), \theta_A(s), B(r), \theta_B(r)) \mid (s,r) \in F_{\delta_1}(A, t(B)) \cap H_{\delta_2}(A, t(B))\}\,. \tag{3.8}$$

The probability for a rigid motion t to be covered by a δ-region corresponding to a random sample pair is then $p_\delta(t) = \frac{|M_\delta(t)|}{|\Omega|} = |F_{\delta_1}(A, t(B)) \cap H_{\delta_2}(A, t(B))|$.

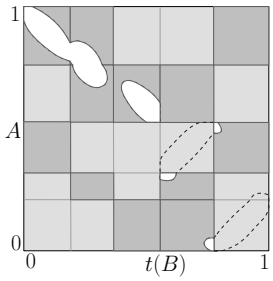


Figure 3.7: Free space diagram corresponding to the two shapes depicted in Figure 3.4 for the case of rigid motions, Approach 2. The cells shaded in dark grey correspond to the segment pairs in the two shapes, for which the direction angles differ by at most δ_2, that is, the free space with respect to H_{δ_2}. The white regions are the intersection of F_{δ_1} and H_{δ_2}.

Clearly, for sets of polygonal curves, or, equivalently, sets of straight line segments, the direction of the tangent line stays the same for all points of one segment, unless we perform some smoothing. Thus, for a segment s_a of the shape A and a segment s_b of B the directions of tangent lines differ by the same amount for all points of the segments, which explains the block structure of H_{δ_2}.

For practical applications it would be reasonable to take an interpolated tangent direction. That way the problem of non-unique tangent line at the corners of the polygon is avoided. And, which is more important, eventual noise is "smoothed off". For example, consider two shapes of which one, the shape B, contains a polygonal chain consisting of short line segments pointing to different directions, and the other, shape A, contains a straight line segment corresponding to the chain in the first shape as depicted in Figure 3.8(a). Then for $\delta_2 < \frac{\pi}{4}$ there is no rigid motion that would match a larger part of the polygonal chain of B to the segment of A, instead, any rigid motion mapping a short edge of the chain in B to a part of the segment in A results in an equally good match, Figure 3.8(b). Although, with sufficient amount of smoothing applied to the direction of tangent line we would get a match that corresponds to our intuitive expectation, Figure 3.8(c).

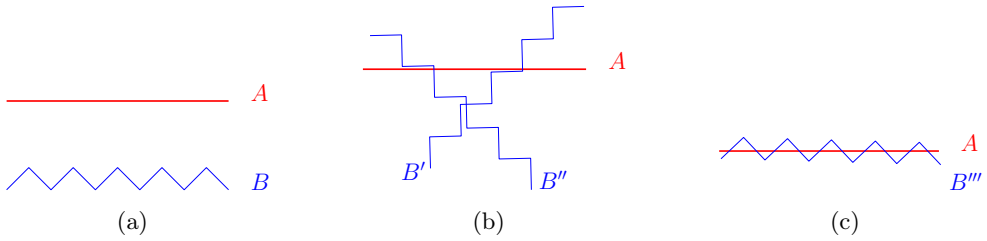


Figure 3.8: Matching under rigid motions with Approach 2: (a) Two shapes, where shape B contains regular noise; (b) Some possible matching positions of B with no tangent line interpolation; (c) Matching position of B if the direction of tangent line is sufficiently smoothed.

The quite simple example in Figure 3.8 can be extended to more complex shapes. The tangent smoothing, or alternatively, curve smoothing prior to matching is especially relevant for noisy data, such as coming from raster images, that already contain a certain amount of noise, or when comparing shapes that were obtained from raster images by different vectorization algorithms.

In the following Lemma we show that for both Approaches the parameter ν in Theorem 3.1 can be eliminated, which completes the proof of part *(iii)* for rigid motions.

Lemma 3.5. *Given two shapes A and B modeled by finite sets of n rectifiable curves in total with total lengths L_A, L_B respectively, the class of the allowed transformations $\mathcal{T}$ as the group of rigid motions, and a tolerance value $\delta > 0$ for Approach 1, or a tolerance tuple (δ_1, δ_2) for Approach 2, let $m = \max(L_A, L_B, n\delta)$. The maximum of $p_\delta(t)$ can be bounded by $\max_{t\in\mathcal{T}} p_\delta(t) \geq \nu$, where $\nu = \frac{\delta^2}{m^2}$ for Approach 1 and for Approach 2 and shapes represented by sets of straight line segments or smooth curves with curvature at most $\frac{\delta_2}{\delta_1}$ $\nu = \frac{\delta_1^2}{m^2}$. For Approach 2 and shapes represented by curves of high curvature $\nu = \frac{\delta_1^2\delta_2^2}{m^2\pi^2}$.*

Proof. For Approach 1 the proof argument is the same as in Lemma 3.4 for translations. We consider curve segments $s_a \subset A$ and $s_b \subset B$ of length δ if they exist, or otherwise the longest curve segments with the lengths $\frac{L_A}{n}, \frac{L_B}{n} \leq \delta$ respectively. Then the rigid motion t_x with rotation angle zero and translation vector mapping the center of s_b to the center of s_a is covered by every δ-region corresponding to a point in s_a and a point in s_b. Therefore, $s_a \times s_b \subset M_\delta(t_x)$ and $p_\delta(t_x) \geq \frac{\delta^2}{m^2}$.

For Approach 2 and if the shapes A and B are represented by straight line segments we again consider the segments s_a, s_b defined as above. Let x_a and x_b denote the centers of s_a and s_b respectively. Observe that the angle θ_a between the tangent line at a point of s_a and the x-axis stays the same for every point of s_a. Similarly, for every point of s_b the angle between the tangent line at that point and the x-axis is θ_b. Then the rigid motion t_x with rotation angle $\alpha = \theta_a - \theta_b$ and

the translation vector $v_x = x_a - \alpha(x_b)$ is covered by every δ-region generated by an arbitrary point in s_a and an arbitrary point in s_b. Thus, $s_a \times s_b \subset M_\delta(t_x)$, and $p_\delta(t_x) \geq \frac{\delta_1^2}{m^2}$.

For shapes A and B represented by sets of smooth curves, if in each of the shapes there exists a part of a curve of length δ such that for every point of that part ($s_a \subset A, s_b \subset B$) the curvature is at most $\frac{\delta_2}{\delta_1}$, then the angles of the tangent lines at every point of s_a or s_b, respectively, differ by at most δ_2. Let $[\theta_{a1}, \theta_{a2}]$ and $[\theta_{b1}, \theta_{b2}]$ denote the intervals of the tangent angles for the points of s_a and s_b respectively. The rigid motion t with rotation angle $\alpha = \frac{1}{2}(\theta_{a2} - \theta_{a1}) - \frac{1}{2}(\theta_{b2} - \theta_{b1})$ and translation vector mapping the center of s_b rotated by α to the center of s_a is contained in every δ-region generated by a point in s_a and a point in s_b. By the same argument as above t maps every point of s_b into a δ_1-neighborhood of every point in s_a. Additionally, the angle of the tangent line at an arbitrary by α rotated point of s_b differs from the angle of the tangent at an arbitrary point of s_b by at most $2\frac{\delta_2}{2}$. An analogous argument holds for the case that in one of the shapes every curve has length less than δ_1 but there exists a curve of length at least L/n, where L is the length of the corresponding shape, such that the curvature at every point of that curve is at most $\frac{\delta_2 n}{L}$. Thus, $s_a \times s_b \subset M_\delta(t_x)$, and $p_\delta(t_x) \geq \frac{\delta_1^2}{m^2}$.

Finally, if there is no part of length δ_1 of any curve of one shape, say A, with curvature at most $\frac{\delta_2}{\delta_1}$ let s_a denote an arbitrary part of a curve in A of length δ_1 if there exists one, or the largest curve of A with length at least $\frac{L_A}{n} \leq \delta_1$ otherwise.

Consider the distribution of angles formed by the tangent line to a point in s_a and the x-axis. If we have a uniform distribution of angles and every angle between 0 and π is present (which is the case if s_a is a circle), then the density function f_a of that distribution is $1/\pi$. For an angle $\theta_a \in [0, \pi)$ the probability of choosing randomly a point of s_a such that the angle of the tangent at that point differs from θ_a by at most $\frac{\delta_2}{2}$ is $\int_{\theta_a - \delta_2/2}^{\theta_a + \delta_2/2} f_a(\varphi)\,\mathrm{d}\,\varphi$. In case of the uniform distribution that probability is $\frac{\delta_2}{\pi}$ for every θ_a. Then by the pigeonhole principle for an arbitrary distribution the maximum $\max_{\theta_a \in [0,\pi)} \int_{\theta_a - \delta_2/2}^{\theta_a + \delta_2/2} f_a(\varphi)\,\mathrm{d}\,\varphi \geq \frac{\delta_2}{\pi}$. Let θ_a denote the angle maximizing that probability over an interval of size δ_2. With probability at least $\frac{|s_a|}{L_A} \cdot \frac{\delta_2}{\pi}$, where $|s_a|$ denotes the length of s_a, a random sample of the shape A chosen by the algorithm contains a point of s_a with the tangent angle in the interval $[\theta_a - \frac{\delta_2}{2}, \theta_a + \frac{\delta_2}{2}]$.

Similarly, if the shape B does not contain a part of length δ_1 with low curvature, let s_b denote an arbitrary part of length δ_1 of a curve in B if there exists one, or the longest curve in B otherwise. Then there is an angle θ_b such that the probability of choosing randomly a point of s_b such that the angle of the tangent at that point differs from θ_b by at most $\frac{\delta_2}{2}$ is greater or equal $\frac{\delta_2}{\pi}$. The probability that a randomly chosen sample of B contains a point of s_b with that property it at least $\frac{|s_b|}{L_B} \cdot \frac{\delta_2}{\pi}$.

Combining these two observations we get that for a rigid motion t with rotation angle $\alpha = \theta_a - \theta_b$ and translation vector that maps the center of s_b rotated by α to the center of s_a the hit probability is $p_\delta(t) \geq \frac{|s_a|}{L_A} \cdot \frac{\delta_2}{\pi} \cdot \frac{|s_b|}{L_B} \cdot \frac{\delta_2}{\pi} \geq \frac{\delta_1^2 \delta_2^2}{m^2 \pi^2}$. □

Discussion: Here we address the differences between the two approaches discussed above.

We already mentioned the problem which can arise in Approach 2 in combination with shapes represented by polygonal curves. If one of the shapes contains a "stairs" kind of noise, it might happen that no pair of points yield a rotation angle that is at least δ_2 close to the one we would expect as the best rotation. All transformations computed by the algorithm get then very few votes, that is, the value of $p_\delta(t)$ for these transformations is small, so the shapes will be rated as not similar.

Approach 1, on the contrary, is insensitive to the "stair"-noise, since it does not take the direction of lines into account and for the example in Figure 3.8 it would give us a matching position depicted in (c). As we have seen, this problem in Approach 2 can be overcome by taking an interpolated direction of the tangent line or by applying some smoothing to the shapes before matching. However, it depends on the relative size of "noise" and interpolation step whether the tangent directions can be sufficiently smoothed. Furthermore, one can easily find examples, some of them are shown in Figure 3.9, where Approach 2 leads to non-intuitive matching results and the problem cannot be fixed by smoothing the tangent directions. Note that matching in all examples was performed with the same distance tolerance value for both approaches. In some of the cases there exists more than one optimal matching position, but only one is depicted in the figure.

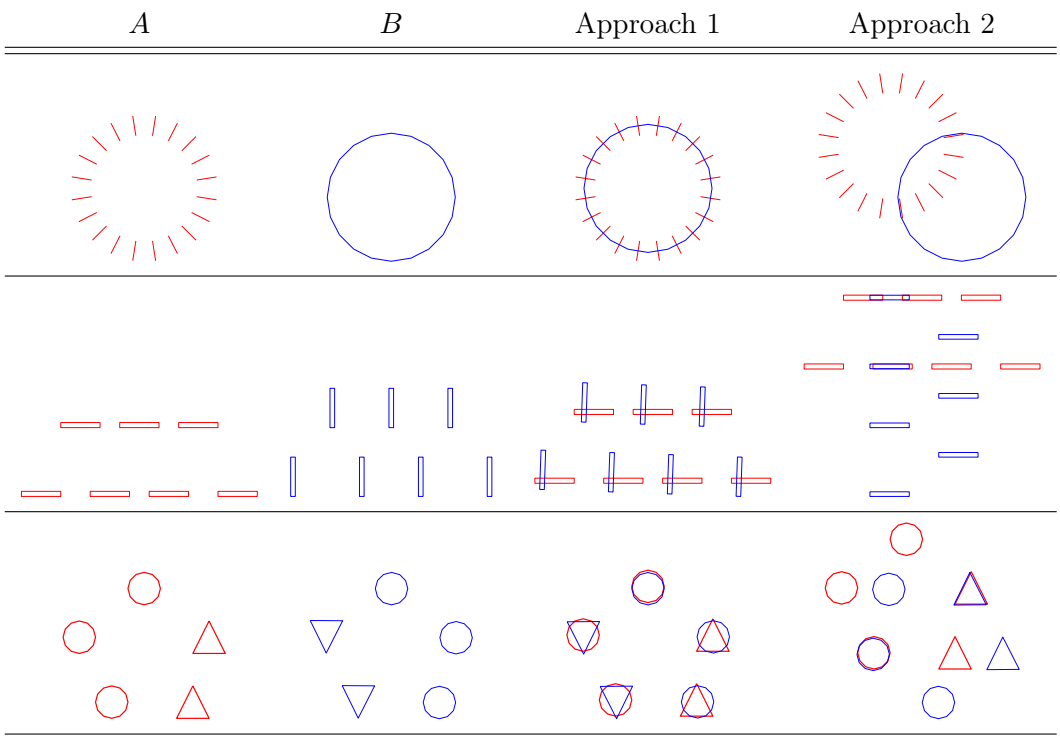
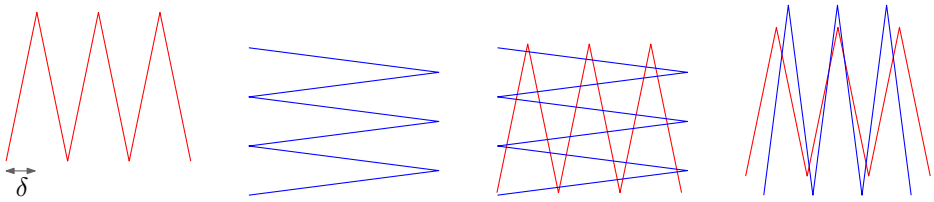


Figure 3.9: Differences in matching results for rigid motions with Approach 1 and 2.

The properties that these examples have in common and that lead to the differences in the matching are the following: At least one of the two compared shapes consists of many rather small components, and thus, no curve smoothing can be applied. Additionally, the perception of the shape is influenced more by the arrangement of the components in the plane, by the so-called spatial layout, than by the shape or the orientation of the individual components.

On the other hand, the direction information can be helpful to achieve better matching results with respect to human perception as the following example shows. Consider two zigzag curves as illustrated in Figure 3.10 and matching with the tolerance value δ such that δ is half of the zigzag width of the shape A. Then, for the purely distance based Approach 1 the transformations with the rotation angles 0 and $\frac{\pi}{2}$ as depicted in Figures 3.10(c) and 3.10(d) respectively are almost equally good. Whereas the matching with Approach 2 uses the direction information and leads to the more intuitive matching of Figure 3.10(d). Note that the example in Figure 3.10 is often used to demonstrate the weakness of the distance based methods as opposed to those that consider orientation, direction, or connectivity information. Thus, the described problem is inherent to the distance based matching and similarity evaluation techniques. Of course, one can argue that a zigzag curve with the zigzag width of at most twice the tolerance value (or any other curve with local feature size at most the tolerance value) can be viewed as a textured region and therefore both depicted matchings are equally good.



(a) shape A (b) shape B (c) Matching with Approach 1 (d) Matching with Approach 2

Figure 3.10: Matching two zigzag curves with and without direction information

It depends on the application whether the direction of curves is important and, therefore, which of the two approaches is appropriate.

The motivation for considering Approach 2 for matching under rigid motion is, that the tangent direction provides additional information about the shape and its orientation. This additional information can then be used to reduce the search space for the optimal transformation. Clearly, if we match two points as well as their tangents, we also match a certain neighborhood of these points. We compared the two approaches experimentally using the MPEG-7 Core Experiment CE-Shape-1 data set, where the shapes are silhouettes of real world objects, and a set of trademark images. In both data sets the curves are nice in a sense that the amount of noise is

small, the curves do not "wiggle" around one point, and the shapes do not contain multiple copies of the same curve segment. In this setting, which we assume to be typical for a practical application, Approach 2 needs much fewer experiments to find a good matching position than Approach 1, and the matchings achieved by both methods do not differ much. These experiments imply that there is a large gap between the theoretically derived bound on the sufficient number of experiments and the actually required number for Approach 2, at least for realistic input data.

Summarizing the above discussion we can say that the direction of tangent lines to the shape provides additional useful information about the shape which helps to reduce the number of experiments and, hence, to reduce the running time of the algorithm. The problem which we encounter with the tangent direction, is that it is an unstable feature for noisy images, and thus, for the shapes extracted from these images. Further, for some images it is not a perceptually relevant feature and might negatively influence the matching results.

We also considered other approaches to matching under rigid motions, two of which we briefly describe here:

Approach 3: Within one random experiment we select, instead of a point and an angle of the tangent line at this point, randomly two points of the shape A, that is $S_A = (a_1, a_2)$, $a_1, a_2 \in A$. From the shape B the first point b_1 is picked randomly, and the second point b_2 is selected randomly from the points of the shape B that have the same Euclidean distance from b_1 as a_2 from a_1, $S_B = (b_1, b_2)$, $b_1, b_2 \in B$ such that $\|b_1 - b_2\| = \|a_1 - a_2\|$. If no such point b_2 exists the sample is discarded and a new one is generated. There exists a unique rigid motion mapping S_B to S_A. The corresponding δ-region is a set of rigid motions that map b_1 into a δ-neighborhood of a_1 and b_2 into that of a_2.

Approach 4: Another possibility is to choose randomly two points in each shape $a_1, a_2 \in A$ and $b_1, b_2 \in B$ and to consider random samples consisting of points $a_1 \in A$ and $b_1 \in B$ and angles θ_a and θ_b defined as an angle between the positive direction of the x-axis and the direction of the vector $\overrightarrow{a_1 a_2}$ or $\overrightarrow{b_1 b_2}$ respectively. The corresponding rigid motion maps b_1 to a_1 and matches the angles θ_b and θ_a. As in Approach 2 the tolerance bound consists of two parameter values, one for the distance between the points and one for the angles. The definition of a δ-region corresponds to that of Approach 2.

In both of the last two methods we aim to reduce the search space for the transformations while avoiding the above mentioned drawbacks of the tangent direction. Approach 3 uses the direction vectors between two arbitrary points of the shape instead of the interpolated direction of the tangent line as in Approach 2. Thus, it is less sensitive to noise and to segmentation of shapes into individual components. However, it introduces additional computational overhead in the process of sam-

ple generation since we first need to find points in shape B that have the specified distance to a point $b_1 \in B$ and then choose randomly one of such points if any exist.

The idea to keep the smoothing properties of Approach 3 but to avoid additional computations was the motivation for looking at Approach 4. In fact, Approaches 3 and 4 lead to similar matching positions in experimental studies. Both methods produce results closer to that of Approach 1 than of Approach 2. As one would expect, in all examples depicted in Figure 3.9 the matching transformation computed with Approaches 3 and 4 correspond to those achieved by Approach 1. Of course, the matching results obtained with Approaches 1, 3, and 4 are not always identical. Figure 3.11 shows an example of two shapes, where Approaches 1, 2, and 3 and 4 produce different results.

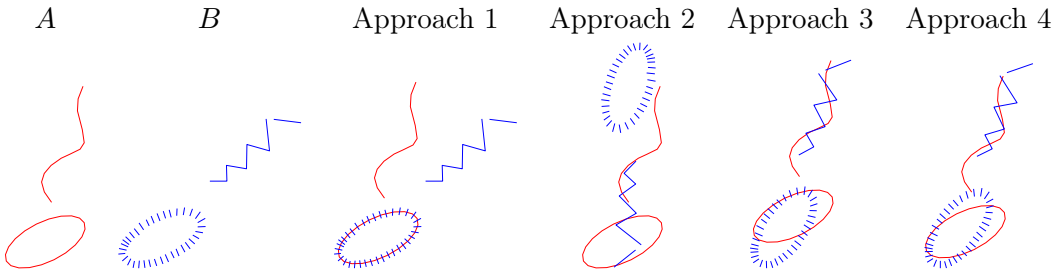


Figure 3.11: Differences in matching results for rigid motions with Approaches 1 to 4.

The difference is due to the fact that Approach 1 does not consider any directional information, and therefore, finds a best match with respect to point proximity. Whereas, in Approaches 3 and 4 the global orientation of the shape influences the direction of the lines connecting random point pairs and, hence, the rotations considered for matching transformations. If the compared shapes have a clear orientation, then Approaches 3 and 4 would find a rigid motion with a rotation angle that matches the orientations well, and for that rotation angle determine a translation that maps the most points of the shapes into the δ-neighborhoods of each other.

Summarizing the above discussion, we can say that Approaches 1 and 2 find matchings emphasizing local correspondences, where Approach 2 is strongly affected by the orientation of line segments comprising the shape, or by the direction of lines tangent to curves, and Approach 1 only considers distances between the points of the shapes. These two approaches have partial-partial matching character. On the other hand, Approaches 3 and 4 match global orientation of shapes and, therefore, have complete-complete matching character.

3.3.3 Similarity Maps

In the case of similarity maps an intuitive approach is to take two random points from each shape as a sample in one experiment, since two pairs of points in the plane

determine uniquely a similarity transformation that maps one pair to the other. The sample space is then $\Omega = A^2 \times B^2$. According to our analysis the similarity map with maximum coverage of the δ-regions is the one maximizing the measure of the set

$$M_\delta(t) = \left\{(a_1, a_2, b_1, b_2) \in A^2 \times B^2 | \text{dist}(t(b_1), a_1) \leq \delta \text{ and } \text{dist}(t(b_2), a_2) \leq \delta\right\} ,$$

which is the set of pairs of point pairs that are at most δ apart. This measure is less intuitive with respect to matching shapes than the ones in the previous cases. The following simple consideration shows, however, that maximizing the Lebesgue measure of $M_\delta(t)$ means also to maximize the Lebesgue measure of the set of point pairs that have distance at most δ, $M'_\delta(t) = \{(a, b) \in A \times B | \text{dist}(t(b), a) \leq \delta\}$. Up to the order of the elements the set $M_\delta(t)$ is exactly the set $M'^2_\delta(t)$. Additionally, in our random experiment we exclude samples where $a_1 = a_2$ or $b_1 = b_2$ because in these cases there is either no similarity transformation that maps one sample to the other, in case $b_1 = b_2$ and $a_1 \neq a_2$, or, in case $a_1 = a_2$, a similarity map is not uniquely defined. Those pairs are, of course, not excluded in the set $M'^2_\delta(t)$, but they make up a subset of dimension six in the eight dimensional space (recall, that we have four points each of dimension two) and have, therefore, Lebesgue measure zero. So, the measure of the set $M_\delta(t)$ is exactly $\left|M'^2_\delta(t)\right| = |M'_\delta(t)|^2$. Since the Lebesgue measure of a set is always non-negative, both functions have maxima at the same values of t.

Note that if $\tilde{p}_\delta(t)$ is a good approximation of $p_\delta(t)$, more precisely, an approximation with a relative error at most ε, then $\tilde{m}_\delta(t) = \sqrt{\tilde{p}_\delta(t)\,|\Omega|}$ is an approximation of $m_\delta(t) = |M'_\delta(t)|$ with a relative error at most ε: Since $\tilde{p}_\delta(t) \geq (1-\varepsilon)p_\delta(t)$ and $\tilde{p}_\delta(t) \leq (1+\varepsilon)p_\delta(t)$ by Lemma 3.1, then

$$\begin{aligned}
\tilde{m}_\delta(t) &= \sqrt{\tilde{p}_\delta(t)\,|\Omega|} \\
&\geq \sqrt{(1-\varepsilon)p_\delta(t)\,|\Omega|} \\
&= \sqrt{(1-\varepsilon)\,|M_\delta(t)|} \\
&= \sqrt{(1-\varepsilon)}m_\delta(t) \\
&\geq (1-\varepsilon)m_\delta(t) && \text{since } 0 \leq 1-\varepsilon \leq 1
\end{aligned}$$

and

$$\begin{aligned}
\tilde{m}_\delta(t) &= \sqrt{\tilde{p}_\delta(t)\,|\Omega|} \\
&\leq \sqrt{(1+\varepsilon)p_\delta(t)\,|\Omega|} \\
&= \sqrt{(1+\varepsilon)}m_\delta(t) \\
&\leq (1+\varepsilon)m_\delta(t) && \text{since } 1+\varepsilon \geq 1
\end{aligned}$$

Therefore, for similarities as well as for translations and rigid motions (with Approach 1) the probabilistic algorithm finds a transformation that approximately

maximizes the measure of the set of point pairs that are in a δ-neighborhood of each other.

Although a standard way to parameterize the space of similarity transformations is by rotation angle α, scaling factor k and a translation vector $v = (v_x, v_y)$ in order to avoid trigonometric functions in the definition of δ-regions it is more convenient to use the parameterization (m_1, m_2, v_x, v_y) where $m_1 = k\cos\alpha$ and $m_2 = -k\sin\alpha$. For general L_p metric a δ-region is then bounded by algebraic surfaces (Inequalities (3.3) or (3.4)), and for the L_1 and L_∞ metrics it is a convex polytope bounded by four pairs of parallel hyperplanes (Inequalities (3.4) for $p = 1$ and (3.5) for $p = \infty$). An arrangement of N convex polytopes in d dimensions can be computed and traversed in time $O(N^d)$, see [23], which is $O(N^4)$ for similarity maps.

Next we show that for similarity transformations there exists a value $\nu > 0$ such that for any two shapes A and B the maximum of the corresponding probability functions is at least ν, which completes the proof of part *(iii)* of Theorem 3.1 for similarities.

Lemma 3.6. *Given two shapes A and B modeled by finite sets of n curves with total lengths L_A, L_B respectively and a tolerance value $\delta > 0$, let $m = \max(L_A, L_B, n\delta)$. If the class of allowed transformations $\mathcal{T}$ is the group of similarity transformations then $\max_{t\in\mathcal{T}} p_\delta(t) \geq \nu$, where $\nu = \frac{\delta^2}{m^2}$.*

Proof. The shape B can always be scaled by the factor $\frac{\delta}{D_B}$, where D_B is the diameter of B, so that the diameter of the scaled shape B is δ. If A contains a connected component of length at least δ, then we can place the scaled shape B in such a way that for any point of a part of A of length δ the distance to any point of the scaled B is at most δ. Therefore, the measure of the set $M_\delta(t)$ for that t is at least $L_B^2 \cdot \delta^2$. The corresponding value of p_δ is $p_\delta(t) = \frac{|M_\delta(t)|}{|\Omega|} \geq \frac{L_B^2 \cdot \delta^2}{L_A^2 L_B^2} = \frac{\delta^2}{L_A^2}$. Otherwise, the largest connected component of A must have length at least $\frac{L_A}{n}$. For the transformation t that maps the scaled B to the largest component of A the measure of $M_\delta(t)$ is then at least $\frac{L_A^2}{n^2} L_B^2$ and $p_\delta(t) \geq \frac{1}{n^2}$. Therefore, $\nu \geq \frac{\delta^2}{m^2}$. □

The Shrinking Problem

The probabilistic algorithm applied to matching under similarity transformations in the described way has what we call a "shrinking problem" which was observed in experiments and can be explained theoretically. Namely, it turns out that, when matching a shape B to a shape A, the transformations that scale the shape B down to some small diameter, e.g., δ, or in the extreme case scale B down to a point, and then map it to some location in the shape A can have about the same similarity value as the transformations that yield an intuitive match.

The reason for this behaviour of the algorithm is the following: As we observed in Lemma 3.6, for a transformation t that scales the shape B down by the factor $\leq \frac{\delta}{D_B}$, where D_B is the diameter of the shape B, and maps the scaled shape B to some location on A the measure of the set $M_\delta(t)$ is approximately $\delta^2 L_B^2$. On

the other hand, if the shapes A and B are congruent, and t_{opt} is a transformation that maps B exactly to A, then for every point b of $t_{\text{opt}}(b)$ there exist some part of the shape A of length *about* δ such that the points of that part of A are in the δ-neighborhood of b. Of course, we cannot determine exactly the measure of the set $M_\delta(t_{\text{opt}})$ since it depends on the regarded shapes, but from the above consideration we can conclude that this measure is close to $\delta^2 L_B^2$. Thus, we get the same estimate of the similarity measure for transformations that scale the shape B down to a point as for the transformation that matches two shapes exactly.

Obviously downscaling one shape to a very small size is not what one intuitively would call a good match. There are several ways to avoid this behaviour, some of which we describe below.

Fixed lower bound for the scaling factor. A very simple possibility to avoid the shrinking problem is to restrict the smallest allowed scaling factor to some value, e.g., $c\frac{\delta}{D_B}$, where c is a constant. This restriction only changes the hit probability function $p_\delta(t)$ for transformations with the scaling factor $\leq c\frac{\delta}{D_B}$ by setting it to zero. In fact, if the best matching transformation for two shapes is the one that scales one shape down to the size of the tolerance bound then we would say that the two shapes do not match well, or are not similar. Thus, by setting a lower bound on the scaling factor we disregard the downsizing transformations and for all other transformations the similarity measure remains unchanged.

Modification of the random sample (Approach 2). Another possibility is to change the random sample taken within one experiment. A natural attribute associated with a point on a curve in the plane which depends on the scaling factor is the curvature. For a plane curve given by Cartesian parametric equations $x = x(s)$ and $y = y(s)$, where $x(s)$ and $y(s)$ are twice continuously differentiable functions, the curvature κ is defined by

$$\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}}$$

and is the inverse of the radius of the curve's osculating circle in the corresponding point. Since we assumed that the curves are given by a constant speed parameterization (as described in Section 2.1), the denominator of the above expression is constant for each curve of the set and can be precomputed at the beginning. The radius r_a of the curve's osculating circle at a point a is the inverse of the absolute value of the curvature at that point and is a real value if the curvature is non-zero.

A random sample consisting of one point a of the shape A, the curvature κ_a of the curve containing a at the point a, and the inclination angle θ_a of the normal to the tangent at the point a and a corresponding random sample (b, κ_b, θ_b) of the shape B uniquely define a similarity map $t = (\alpha, k, v)$ such that in the shape $t(B)$ the absolute value of curvature at the point $t(b)$ is equal to $|\kappa_a|$ ($k = \frac{|\kappa_b|}{|\kappa_a|}$), the inclination angle of the normal to the line tangent to $t(b)$ is θ_a ($\alpha = \theta_a - \theta_b$), and the point b

is mapped to the point a ($v = a - M_{k\alpha}b$), where $M_{k\alpha} = \begin{pmatrix} k\cos\alpha & -k\sin\alpha \\ k\sin\alpha & k\cos\alpha \end{pmatrix}$. Similarly as in Approach 2 for the case of rigid motions with every random sample pair $S_A = (a, \kappa_a, \theta_a), S_B = (b, \kappa_b, \theta_b)$ we also add the sample pair $S_A = (a, \kappa_a, \theta_a), S_B = (b, \kappa_b, \theta'_b)$, where $\theta'_b = \theta_b + \pi$ since we consider the curves to be undirected.

This approach is not directly applicable to shapes represented by polygonal curves, since the curvature of a straight line segment is zero and the curvature at the vertices of a polygonal curve is infinite. Therefore, some curvature approximation is needed. There are some methods for curvature approximation of discrete curves in the pattern recognition field, an extensive survey is included in the Master's theses by Driemel [19] and the article by Hermann and Klette [36] gives a short overview. Alternatively, each polygonal curve can be approximated by a smooth curve, or a constant number of smooth curves, for example as a B-spline curve, or as a set of circular arcs or biarcs, see e.g., the approximation algorithm by Drysdale et al. [20]. The advantage of the latter method is that each circular arc can easily be parameterized and has constant curvature. However, a polygonal curve is typically itself an approximation of some curve sampled by a finite number of points, for example a GPS track, a curve in a pixel image, or an outline of a real world object registered with a laser range scanner. Thus, whether a further approximation is desired or appropriate depends on the origin of data and the application.

We already discussed the possibilities of interpolating the direction of the tangent line at a point or smoothing a curve prior to sampling in the description of Approach 2 for rigid motions. Of course it is reasonable, for the sake of consistency, to apply related interpolation or smoothing methods for the curvature computation and for computation of the tangent direction.

For this approach the user needs to specify a triple of tolerance values $(\delta_1, \delta_2, \delta_3)$ where δ_1 denotes the maximal distance for points to be considered close, δ_2 the maximal allowed difference of the angles formed by the tangents at the points, and δ_3 the maximal allowed relative curvature difference. Then the δ-region corresponding to the sample pair S_A, S_B is, by definition, the set of all similarity transformations t, such that the transformed point $t(b)$ has distance at most δ_1 from the point a, the angle of inclination of the normal of the tangent at the point $t(b)$ in the transformed shape B differs from θ_a by at most δ_2, and the relative deviation of the absolute value of the curvature at the image point of b within the transformed shape B, denoted by $t(\kappa_b)$ from the absolute value of the curvature at the point a is at most δ_3. The latter condition is expressed by

$$\frac{|t(|\kappa_b|) - |\kappa_a||}{|\kappa_a|} \leq \delta_3.$$

We decided to use relative tolerance for the scaling factor instead of absolute tolerances as in the case of distance and tangent directions, since the perceived change of the shape caused by the scaling is relative to the start size of the shape and not absolute in the change of the scaling factor.

Since translation and rotation do not change curvature, the curvature at the point b after applying t to the corresponding curve is $t(\kappa_b) = \kappa_b/k$, where k is the scaling factor of t. Then, the curvature tolerance condition implies the following two linear inequalities:

$$\frac{|\kappa_b|}{(1+\delta_3)\,|\kappa_a|} \leq k \leq \frac{|\kappa_b|}{(1-\delta_3)\,|\kappa_a|} \tag{3.9}$$

if $\kappa_a, \kappa_b \neq 0$. A drawback of this approach is that it cannot handle cases where one of the curvature values is zero, since in that case there is no scaling factor that would map one curvature value to the other. For smooth curves, if there are finitely many points with curvature value zero, we can split the curves at the points of zero curvature and treat the curve segments separately. For smooth curves containing segments of zero curvature we need some "special treatment", for example, we can use some small value ε for the curvature instead of zero. Further, we assume that there are no points with infinite curvature values, since the curve segments can be split at sharp corners and the new segments treated separately. For polygonal curves, where the only curvature values are zero and infinity, for this approach we need some curvature approximation procedure as described above.

Recall that we consider the parameterization of the similarity space given by (m_1, m_2, v_x, v_y), where $m_1 = k\cos\alpha, m_2 = -k\sin\alpha$. In this parameterization the scaling factor k is the square root of $m_1^2 + m_2^2$. Since we do not include reflections in the class of similarity maps, the scaling factor k is positive and, therefore, the equations of the type $k \leq c$ and $k \geq c$ are equivalent to equations $k^2 \geq c^2$ and $k^2 \geq c^2$ respectively. Condition (3.9) results thus in two non-linear constraints:

$$\begin{aligned} m_1^2 + m_2^2 &\leq \left(\frac{\kappa_b}{(1-\delta_3)\kappa_a}\right)^2 \\ m_1^2 + m_2^2 &\geq \left(\frac{\kappa_b}{(1+\delta_3)\kappa_a}\right)^2 \end{aligned} \tag{3.10}$$

The tangent direction condition $\alpha \in I = [\theta_a - \theta_b - \delta_2, \theta_a - \theta_b + \delta_2]$ imposes the following constraints on parameters m_1 and m_2:

$$\begin{aligned} m_1 &\leq \left(\max_{\alpha' \in I} \cos\alpha'\right) \sqrt{m_1^2 + m_2^2} \\ m_1 &\geq \left(\min_{\alpha' \in I} \cos\alpha'\right) \sqrt{m_1^2 + m_2^2} \\ m_2 &\leq \left(\max_{\alpha' \in I} \sin\alpha'\right) \sqrt{m_1^2 + m_2^2} \\ m_2 &\geq \left(\min_{\alpha' \in I} \sin\alpha'\right) \sqrt{m_1^2 + m_2^2}\ . \end{aligned}$$

Similar to Approach 2 in the case of rigid motions, we do not compute the angles explicitly. Instead, we assume that the tangent lines are given in Hesse normal form,

which gives us directly the values of sine and cosine of the angles of inclination of the normals. Thus, the minimum and maximum values of sine and cosine in the interval I can be computed without applying trigonometric functions as described in Section 3.3.2 on page 42.

The square root function is avoided by squaring each side of the inequality and adding additional constraints in order to preserve the sign information. We explain the modification for the first two constraints on m_1, the second pair is to be handled analogously. Let M_c denote $\max_{\alpha' \in I} \cos \alpha'$ and m_c denote $\min_{\alpha' \in I} \cos \alpha'$ for shortness of notation. There are three possibilities for the sign of M_c and m_c: (a) both are non-negative, (b) both are non-positive, and (c) the signs are different. In the first case (a) the inequalities are squared and the additional constraint $m_1 \geq 0$ is added:

$$\begin{aligned} m_1^2 &\leq M_c^2 \cdot (m_1^2 + m_2^2) \\ m_1^2 &\geq m_c^2 \cdot (m_1^2 + m_2^2) \\ m_1 &\geq 0 \ . \end{aligned}$$

In the second case (b) the inequality signs have to be reversed:

$$\begin{aligned} m_1^2 &\geq M_c^2 \cdot (m_1^2 + m_2^2) \\ m_1^2 &\leq m_c^2 \cdot (m_1^2 + m_2^2) \\ m_1 &\leq 0 \ . \end{aligned}$$

In the third case the interval $[m_c, M_c]$ is split into two $[m_c, 0]$ and $[0, M_c]$, so that the sign of m_1 stays the same within each interval. Then for each of the intervals a δ-region is defined as described above.

Note that if both intervals for sine and cosine contain zero, than for one sample we need to generate four δ-regions. Although for sufficiently small values of δ_2 ($\delta_2 < \frac{\pi}{4}$) only one of the intervals may contain zero.

Finally, the distance condition is expressed by Inequalities (3.3) to (3.5) with the linear transformation matrix $M = \begin{pmatrix} m_1 & m_2 \\ -m_2 & m_1 \end{pmatrix}$ and tolerance value δ_1. Thus, a δ-region is a semi-algebraic set defined by nine or twelve non-linear inequalities.

Vote modification. Another possibility to avoid the shrinking problem is to generate random samples S_A, S_B and the corresponding δ-regions as in Approach 1, but to give each δ-region a weight which depends on the scaling factor of the transformation mapping S_B to S_A. For example, instead of counting one for each vote we could assign to a transformation with the scaling factor s the weight s, or the weight $1 + \lambda s$ for some constant $\lambda > 0$. Then the weight of a cell of the arrangement of δ-regions is the sum of the weights of the regions covering that cell. The transformations of the cell with the highest weight are considered to be the transformations yielding the best match. Since the weights assigned to votes do not affect the random experiments generating the votes, the number of δ-regions covering a cell of the

arrangement divided by the total number of votes still gives an approximation of the p_δ function, which can be used for rating the similarity of the shapes. This way we avoid the transformations with high values of p_δ and small scaling factor that cause the shrinking problem.

This heuristic approach is motivated by the analysis of the density function of the scaling factor distribution for some simple geometric objects in Section 4.3 and the experiments on the MPEG-7 shape B dataset described in Section A.2. We consider an experiment where the samples S_A, S_B consist of two points each, just as in the case of similarity maps, and examine the probability distribution of the random variable $X = \frac{d_A}{d_B}$, which corresponds to the scaling factor, where d_A, d_B are the distances between the points in S_A and S_B respectively. It turns out that the density function $f(x)$ of that distribution for two straight line segments has its maximum at zero (see Section 4.3.2), and the experimental evaluation for the shapes in the MPEG-7 dataset also shows that the density function attains its maximum value for some values of x close to zero independent of the size of the compared shapes (Section A.2). In fact, the function $g(x) = \lambda x \cdot f(x)$, where $\lambda > 0$ is a constant, actually attains its maximum at the value $x = \frac{L_A}{L_B}$ in case A and B are straight line segments or circles, where L_A, L_B are the lengths of the corresponding segments. The formal analysis can be found in Section 4.3.2 for the case of line segments and in Section 4.3.4 for circles. Furthermore, for the MPEG-7 dataset the maximum of the function $g(x) = \lambda x \cdot f(x)$ with $\lambda = 1$ was attained at the value of x that corresponds to the *expected* scaling factor as reported in Section A.2. By *expected* factor we mean the scaling factor that makes two compared shapes approximately equally large. Since the shapes do not match exactly, the optimal scaling factor is hard to determine.

Applying this modification directly to matching under similarity transformations by assigning the scaling factor as the weight for the δ-regions does not lead to good matching results, since the votes for the same scaling value in the scaling factor experiment are spread in the four dimensional space of similarity transformations. Testing the vote modification strategy experimentally with the shapes from the MPEG-7 dataset we found that assigning a weight $1 + \lambda x$, where $\lambda > 0$ is a constant, to the δ-region corresponding to a sample pair S_A, S_B where x is the scaling factor of the transformation mapping S_B to S_A results in a most intuitive matching transformation. Setting $\lambda = 1$ already results in good matching positions of the shapes. This weighting method accounts for the number of δ-regions containing a transformation, and penalizes small scaling factors by giving the corresponding δ-regions smaller weights. Thus, if two transformations have the same value of $p_\delta(t)$ the one with the larger scaling factor will receive larger weight. By adjusting the value of the constant λ we can control the influence of the scaling factor on the matching results.

It may also be reasonable to either bound the largest possible scaling factor, or to consider only the cells of the arrangement that are covered at least by a certain smallest amount of δ-regions. Otherwise it might happen that a single δ-region

corresponding to some very large scaling factor has the highest weight at the end of the experiment. The weight modification method with this additional restriction worked well in the experiments on the shapes of the MPEG-7 dataset.

3.3.4 Homotheties

Homothety is defined as a similarity transformation which preserves orientation. That means, a homothety transformation $t = (k, v_x, v_y)$ consists of a scaling factor k and a translation vector $v = (v_x, v_y)$ and maps a point b to a point $kb + v$. The corresponding linear transformation matrix M defined by Equation (3.2) is in this case

$$M = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix},$$

i.e., $m_1 = m_4 = k$, $m_2 = m_3 = 0$. The transformation space $\mathcal{T}$ is three dimensional, $\mathcal{T} = \mathbb{R}^+ \times \mathbb{R}^2$.

Similar as in the case of rigid motions, for two points a and b in the plane there exists no unique transformation that maps b to a. Rather for a every scaling factor k we can find a unique translation vector $v = a - kb$ such that the resulting homothety transformation performs the desired mapping. Therefore, the set of transformations that map one of the points to the other is one dimensional and forms a straight line in the three dimensional transformation space. While a sample consisting of one point of each shape is not sufficient to determine a single homothety transformation mapping the sample point of one shape that of the other, a sample consisting of two points overdetermines the transformation, in a sense that in general there may not exist a homothety which maps two points of one shape to arbitrary two points of the other. In the following some approaches to sample generation and the corresponding matching results achieved by the probabilistic algorithm are discussed.

Approach 1. A sample $S_A = (a, \kappa_a)$ of shape A and $S_B = (b, \kappa_b)$ of B taken within a single random experiment in this case consists of a point $a \in A$ and $b \in B$ selected randomly under the uniform distribution with respect to length, and the corresponding curvature values κ_a and κ_b of the curves in A and B containing the points a and b. The sample space Ω is then a subset of $\mathbb{R}^2 \times A \times B$. The homothety transformation $t' = (k', v'_x, v'_y)$ that maps S_B to S_A is defined as $k' = \frac{\kappa_b}{\kappa_a}$ and $v' = a - kb$, if $\kappa_a \neq 0$.

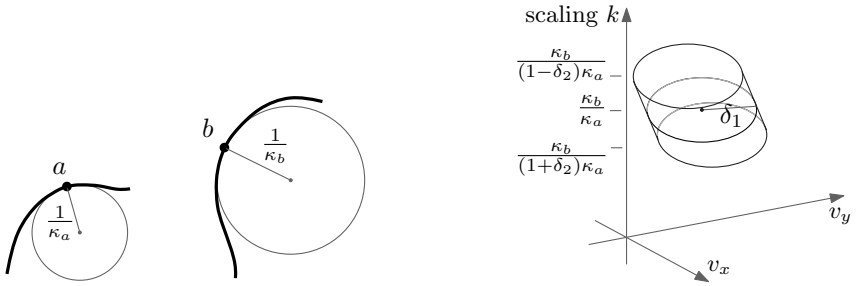
Since the components of the samples are of different nature (points and curvature), it is reasonable to have two different values for the tolerance bound: $\delta = (\delta_1, \delta_2)$, where δ_1 denotes the maximal distance the points may be apart to be considered close and δ_2 controls the maximally allowed tolerance in the curvature. Furthermore, we will consider δ_2 as the relative error bound for the curvature and require that $0 < \delta_2 < 1$. Then the δ-region corresponding to the sample pair S_A, S_B is, by definition, the set of all homothety transformations t, such that the transformed

point $t(b)$ has distance at most δ_1 from the point a, and the relative deviation of the curvature at the image point of b within the transformed shape B, denoted by $t(\kappa_b)$, from the curvature at the point a is at most δ_2. The curvature tolerance condition implies the following two linear inequalities:

$$k \leq \frac{\kappa_b}{(1-\delta_2)\kappa_a} \qquad k \geq \frac{\kappa_b}{(1+\delta_2)\kappa_a} \tag{3.11}$$

if $\kappa_a, \kappa_b \neq 0$. Similarly to Approach 2 for similarity transformations, this approach has a problem with the cases where one of the curvature values is zero, since in that case there is no real values scaling factor that would map one curvature value to the other. As mentioned in the previous section we can either assume that the shapes A and B do not contain curve pieces with zero curvature, or use some small value ε for the curvature instead of zero. We can also assume that there are no points with infinite curvature values, since the curve segments can be split at sharp corners and the new segments treated separately.

The distance condition is expressed by Inequalities (3.3) to (3.5) with the linear transformation matrix M as defined above and tolerance value δ_1. Thus, a δ-region is a semi-algebraic set defined by three or six inequalities of degree at most p, where p depends on the chosen distance metric L_p, or of degree one in case of L_∞. Such δ-region is bounded in scaling dimension by two planes $k = \frac{\kappa_b}{(1-\delta_2)\kappa_a}$ and $k = \frac{\kappa_b}{(1+\delta_2)\kappa_a}$ and for every fixed k the cross-section parallel to the translation plane is exactly a δ_1-neighborhood of the translation mapping the point kb to the point a. That is, a δ-region is a shifted cylinder as illustrated in Figure 3.12.



(a) Sample points a and b with corresponding curvatures κ_a and κ_b.

(b) A δ-region in the three dimensional transformation space corresponding to a sample pair.

Figure 3.12: A δ-region in the space of homothety transformations with Approach 1.

With this approach the algorithm finds a homothety transformation t that maximizes the measure of the set $M_{\delta_1,\delta_2}(t)$ of point pairs with distance at most δ_1 after the transformation t is applied to shape B and the curvature at these points is similar, where similar means that the relative difference of the curvature values is at most

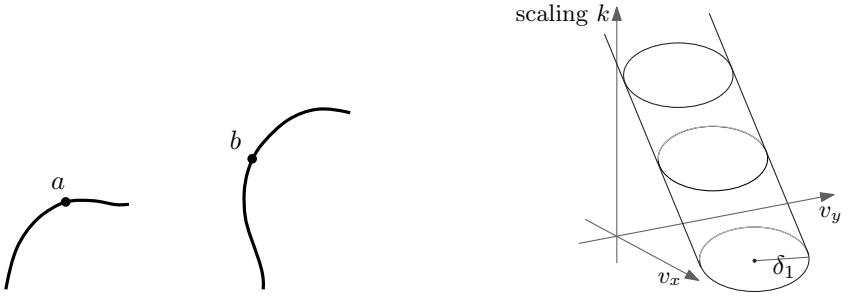
δ_2. Formally:

$$M_{\delta_1,\delta_2}(t) = \left\{(a,b) \in A \times B | t(b) \in U_{\delta_1}(a) \text{ and } \frac{|t(\kappa_b) - \kappa_a|}{\kappa_a} \leq \delta_2\right\} \quad .$$

Since this definition of a good match includes, beside the closeness of the shape points, the similarity of the curvature values, and the curvature is a local feature, Approach 1 favours the transformations that map common local features of two shapes to each other.

Note that this approach as described above is not directly applicable to matching shapes modeled by polygonal curves, since the only curvature values that occur at any point of a polygonal curve are either zero or infinity. As mentioned in the description of similarity transformations, there are curvature approximation methods in the field of pattern recognition which can be applied to approximate the curvature of a polygonal curve, see [19, 36] for surveys. Alternatively, a polygonal curve can be approximated by a smooth or a piecewise smooth curve prior to the sample generation.

Approach 2. The second approach is to use a single random point of each shape $a \in A$ and $b \in B$ as a sample in one random experiment and to define a δ-region as a set of homothety transformations that map the point b into the δ-neighborhood of the point a. The sample space is then $\Omega = A \times B$. A δ-region is a semi-algebraic set defined by the Inequalities (3.3), (3.4) or (3.5), depending on the underlying distance metric L_p. The δ-region corresponding to a sample pair $a \in A,\ b \in B$ is an unbounded shifted cylinder as illustrated in Figure 3.13.



(a) A random sample consisting of a single point of each shape: $a \in A$ and $b \in B$.

(b) A δ-region in the three dimensional transformation space corresponding to the sample pair (a, b).

Figure 3.13: A δ-region in the space of homothety transformations with Approach 2.

As mentioned in Section 3.2.4, in this case the algorithm finds a transformation t that maximizes the measure of the set $M_\delta(t)$ of point pairs that have distance at most δ after the matching transformation is applied to the shape B:

$$M_\delta(t) = \{(a,b) \in A \times B | t(b) \in U_\delta(a)\} \quad .$$

For the completeness of the relative error bound proof we also need to show that there exists a value ν such that the maximum of the probability function $p_\delta(t)$ is always at least ν. Note that in the proof of Lemma 3.6, which states the existence of such value ν for the class of similarity transformation, the rotation component of the similarity transformations was never used. Therefore, the statement of the Lemma holds also for the class of homothety transformations in combination with Approach 2.

The next two approaches are heuristics aimed to reduce the search space compared to that of Approach 2 but to avoid the sensitivity to local features, and thus sensitivity to noise, specific to Approach 1.

Approach 3. A random sample within one experiment is defined as a pair of random points of the shape A, $S_A = (a_1, a_2) \subset A$, and a pair of points of the shape B, $S_B = (b_1, b_2) \subset B$, where the point b_1 is chosen randomly and the point b_2 is a point that lies on a straight line l, which goes through b_1 and is parallel to the line through a_1, a_2. If the intersection of the line l with the shape B contains more than one point, a sample point is chosen randomly among those. If no such point exists the sample is discarded and a new one is generated. There exists a unique homothety transformation that maps b_1 to a_1 and b_2 to a_2. The corresponding δ-region is defined as a set of transformations that map the point b_1 into a δ-neighborhood of the point a_1 and b_2 into that of a_2. A δ-region defined this way is an intersection of two δ-regions defined in Approach 2 corresponding to the point pairs a_1, b_1 and a_2, b_2.

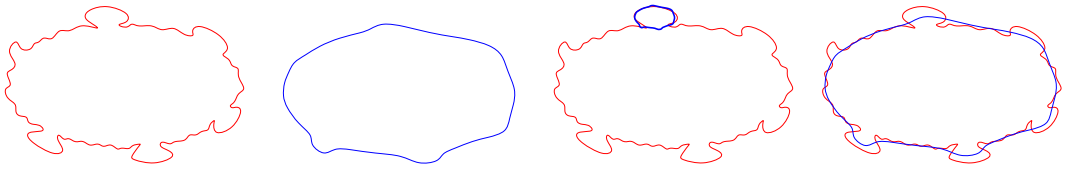
The underlying measure of similarity for this approach corresponds to the measure maximized by Approach 1 in the case of similarity maps, that is, the measure of the set of pairs of point pairs that are at most δ apart is maximized. As mentioned in the description of the similarity transformation, maximizing the measure of this set is equivalent to maximizing the measure of the point pairs of two shapes that are mapped into the δ-neighborhood of each other.

Approach 4. A random sample in this approach consists of a point randomly selected from each shape, $a \in A$ and $b \in B$, and a positive real number associated with that point, d_a and d_b respectively, which denotes the distance to another randomly selected point from the same shape, that is, $S_=(a, d_a)$ and $S_B = (b, d_b)$. The transformation that maps S_B to S_A is then $t = (k, v)$ with $k = \frac{d_a}{d_b}$ and $v = a - kb$. The corresponding δ-region can be defined similarly to that in Approach 1, using two tolerance values δ_1, δ_2, where δ_1 is used to specify the distance conditions given by Inequalities (3.3) to (3.5), and δ_2 expresses the maximally allowed relative difference of the distances d_a and $t(d_b)$. The latter condition is expressed by Inequalities (3.11) where $\frac{\kappa_b}{\kappa_a}$ is replaced by $\frac{d_a}{d_b}$.

The underlying similarity measure in this case is not as clearly defined as for the previously describes methods. Informally, Approach 4 finds a homothety transformation maximizing the measure of the set of point pairs that are δ close to each

other while scaling the shape A approximately to the size of the shape B.

Discussion. Approach 1 uses curvature to limit the range of the scaling factor in the definition of a δ-region. Although the curvature is a natural parameter associated with curves which depends on the scaling factor, it is also a purely local feature. Therefore, it can be very sensitive to noise if the data comes from some sampling procedure. Additionally, Approach 1 can result in a non-intuitive match if the two shapes are defined with different detail levels, as illustrated in Figure 3.14. Whereas Approach 2, which is only based on the distances between the points of the shapes leads to more intuitive matching results, see Figure 3.14(d), but has a larger search space.



(a) Shape A (b) Shape B (c) Matching with Approach 1 (d) Matching with Approach 2

Figure 3.14: Matching two shapes under homothety transformations with Approach 1 and Approach 2.

Approaches 3 and 4 attempt to combine the advantages while avoiding the drawbacks of the first two. In Approach 3 a δ-region corresponding to a sample pair in one experiment covers the transformations that map two points of the shape B into δ-neighborhoods of two points of the shape A, thus the definition of a good match is based only on distances and the size of a δ-region is considerably smaller than that in Approach 1. On the other hand, generating a random sample of the shape B has additional computational overhead, since we first need to compute all intersection points of the shape B with a given line and then select one of these points randomly.

Approach 4 avoids the computational overhead of Approach 3 but restricts the search space for the optimal transformation in a similar way. The range of the scaling factor associated with a sample pair reflects the ratio of the distances between the points in two shapes, which is a global feature, in contrast to the local feature – curvature, in the first approach. It also means that the range of the scaling factors that match the average distance between two points in each of the shapes have higher probability to be covered by a δ-region. Note that the average distance between two points is not necessarily related to the diameter of the shape, and, therefore, the scaling factor matching the average distances is not necessarily the same as the one that matches diameters.

Summarizing the above discussion, we can say that Approaches 3 and 4 tend to scale the shape B to be of roughly the same size as the shape A and with that scaling

find the best translation to match the shapes with respect to distances, and therefore have complete matching character. Approaches 1 and 2 find a transformation that maps the largest similar parts of the shapes close to each other, where the similarity of the parts is defined by local parameters: curvature and point proximity in the first case, and only point proximity in the second. Therefore, these approaches have a partial matching character. Figure 3.15 gives an example of differences in matching results depending on the sample selection strategy.

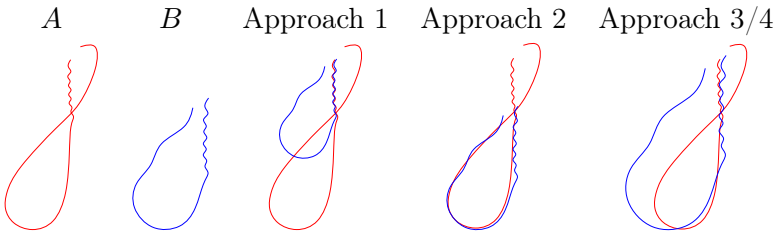


Figure 3.15: Differences in matching results for homothety transformations with Approaches 1 to 4.

3.3.5 Shear Transformations

A shear is an affine transformation in which all points along a given line l remain fixed while other points are shifted parallel to l by a distance proportional to their perpendicular distance from l, as described in Section 2.2.

In the following we will describe the details of the probabilistic algorithm for the transformations consisting of horizontal shear and translation, vertical shear can be handled analogously. The transformation space $\mathcal{T}$ is then three dimensional $\mathcal{T} = \mathbb{R}^3$. A transformation $t \in \mathcal{T}$ is described by three parameters (m, v_x, v_y) and maps a point p to the point $M \cdot p + v$, where

$$M = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix}$$

is the shear transformation matrix, and $v = (v_x, v_y)$ is the translation vector.

For two points a and b there is no unique transformation that maps b to a, rather for every shear parameter m there exists a unique translation vector that completes the desired transformation. For a pair of two points a_1, a_2 nd b_1, b_2, in general, there exists no transformation that maps b_1 to a_1 and b_2 to a_2. Therefore, similar to the case of rigid motions and homothety transformations, there are several possibilities to define a random sample and, correspondingly, a δ-region in the transformation space.

Approach 1. The first approach results from the observation that a shear transformation changes the slopes of the lines, and, as mentioned in the Section 3.3.2,

that a natural slope information associated with a point on a curve is the slope of the tangent line. A random sample in this case consists of randomly selected points $a \in A$ and $b \in B$, and an angle θ_a and θ_b respectively between the tangent line at that point and the x-axis, i.e., $S_A = (a, \theta_a)$, $S_B = (b, \theta_b)$. Note that the orientation of the tangent lines is not a part of the shape information, and is therefore discarded, so the angles θ_a, θ_b lie in the interval $[0, \pi)$. The sample space Ω is then a subspace of $A \times B \times [0, \pi)^2$.

A sample pair $S_A = (a, \theta_a), S_B = (b, \theta_b)$ define uniquely a transformation $t = (m, v_x, v_y)$ that maps b to a and the angle defined by the tangent line to the transformed shape B at the image of point b is exactly θ_a, for $\theta_a, \theta_b \neq 0$. Obviously, there is is no transformation mapping S_B to S_A if $\theta_b = 0$ and $\theta_a \neq 0$, since shear along x-axis does not change the slope of horizontal lines. Also if $\theta_a = 0$ and $\theta_b \neq 0$, which means that the tangent line at a is horizontal and the tangent at b is not, there exists no finite value of m that yields the desired transformation. On the other hand, if both $\theta_a, \theta_b = 0$ then, since every shear factor maps a horizontal line to itself, the transformation is not unique. For $\theta_a, \theta_b \neq 0$ the shear parameter m is computed as $m = \frac{1}{\tan\theta_a} - \frac{1}{\tan\theta_b}$ and the corresponding translation vector is $v = a - M_{\text{hor.shear}} b$, if $\theta_a, \theta_b \neq \frac{\pi}{2}$. If both tangents are vertical, the shear parameter is $m = 0$. For $\theta_b = 0,\ \theta_a \neq 0$ the value of m is $1/\tan\theta_a$, and for $\theta_a = 0,\ \theta_b \neq 0$ the shear is $m = -1/\tan\theta_b$.

Due to heterogeneity of the sample data it is once again reasonable to use two different values $\delta = (\delta_1, \delta_2)$ for the tolerance bounds in the definition of a δ-region. A δ-region corresponding to a sample pair S_A, S_B is defined as the set of transformations that map the point b into a δ_1-neighborhood of the point a so that the angle formed by the tangent line to the image of point b differs from θ_a by at most δ_2. A transformation covered by the most δ-regions is then a one approximately maximizing the measure of the set $M_\delta(t)$ defined by Equation (3.8) in the description of Approach 1 for rigid motion in Section 3.3.2.

The distance condition for the points in a sample is, again, expressed by the inequalities (3.3) to (3.5), depending on the underlying metric L_p, with Matrix $M_{\text{hor.shear}}$ as the transformation matrix and δ_1 for the tolerance value.

The inequalities expressing the similarity of tangent slopes can be derived in the following way: Consider a line l which forms an angle θ with the x-axis as depicted in Figure 3.16, l is defined by the equation $y = \frac{1}{\cot\theta} x + q$ for some constant q. The image of the line l under the shear transformation with parameter m is then a line l' with equation $y = \frac{x}{\cot\theta + m} - \frac{mq}{\cot\theta + m} + q$, were $\cot\theta + m$ is the cotangent of the new angle θ', i.e., $\cot\theta' = \cot\theta + m$. Note that the last observation holds also for $\theta = \frac{\pi}{2}$, since a vertical line is mapped to a line with a slope $1/m$.

A condition $\theta' \in [\theta_1, \theta_2] \subset (0, \pi)$ is then equivalent to the condition

$$\cot\theta_2 \leq \cot\theta + m \leq \cot\theta_1 \ ,$$

since cotangent is a continuous monotone decreasing function on the interval $(0, \pi)$. With this observation we can conclude that for a random sample with $\theta_b \neq 0$ and $\theta_a \in$

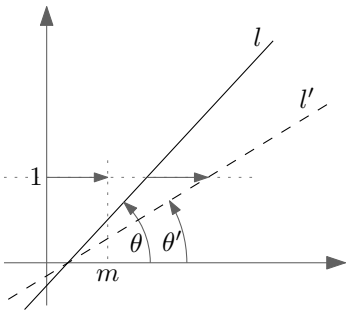


Figure 3.16: Image of the line $l : y = \frac{1}{\cot\theta}x + q$ under shear transformation with parameter m is the line $l' : y = \frac{x}{\cot\theta+m} - \frac{mq}{\cot\theta+m} + q$.

$(\delta_2, \pi - \delta_2)$ the definition of a δ-region is extended by the following two inequalities:

$$m \geq \cot(\theta_a + \delta_2) - \cot\theta_b \tag{3.12}$$

$$m \leq \cot(\theta_a - \delta_2) - \cot\theta_b \tag{3.13}$$

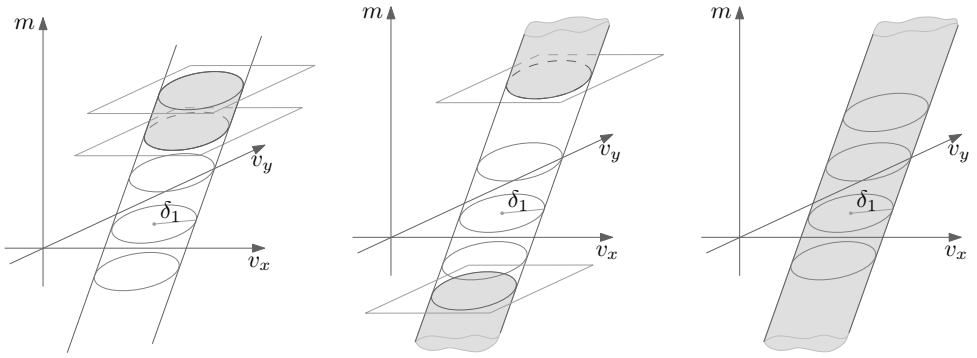
where all inequalities in the definition of a δ-region are logically combined by an AND-operator, Fig 3.17(a).

In case $\theta_b \neq 0$ and $\theta_a \in [0, \delta_2) \cup (\pi - \delta_2, \pi)$ the ranges of the shear factor defined by the inequalities (3.12) and (3.13) are disjoint. The corresponding δ-region consists of two parts: one is defined by the distance constraints and inequality (3.12), and the other by distance constraints and inequality (3.13), Fig 3.17(b).

If $\theta_b = 0$ and $\theta_a \notin [0, \delta_2] \cup [\pi - \delta_2, \pi)$ then there exists no shearing that maps a line with a declination angle θ_b to a line with a declination angle of $\theta_a \pm \delta_2$. The sample pair has to be discarded. For $\theta_b = 0$ and $\theta_a \in [0, \delta_2] \cup [\pi - \delta_2, \pi)$ every shear transformation performs the desired mapping. Therefore the factor m is unbounded and the corresponding δ-region is defined only by the distance constraints, Fig. 3.17(c).

Similar to the case of rigid motions, we do not compute the angles θ_a and θ_b explicitly, but assume that the tangent lines are given in Hesse standard form, which contains the sine and cosine values of the inclination angle of the normal to the tangent line. Additionally, we assume that the tolerance value δ_2 is given by its cosine value. Then using simple trigonometric rules the computation of the cotangent values in Inequalities (3.12) and (3.13) includes only algebraic operation on the parameters of the tangent lines and the tolerance value.

Approach 2. As mentioned in the description of the general approach, Section 3.2.4, we can always use a single random point of each shape as a sample within one experiment. That is, $S_A = a \in A$ and $S_B = b \in B$, the sample space is $\Omega = A \times B$. The corresponding δ-region is then defined only by the point distance constraints (3.3), (3.4), or (3.5), and has the shape of a shifted cylinder as depicted in Figure 3.17(c).



(a) (b) (c)

Figure 3.17: The distance constraints define an infinite shifted cylinder in the parameter space of transformations. For the sample pairs with $\theta_b \neq 0$ and $\theta_a \in (\delta_2, \pi - \delta_2)$ the δ-region is a part of this cylinder bounded by two planes (a). If $\theta_b \neq 0$ and $\theta_a \in [0, \delta_2) \cup (\pi - \delta_2, \pi)$ the corresponding δ-region consists of two parts, each of which is bounded on one side by a plane (b). For $\theta_b = 0$ and $\theta_a \in [0, \delta_2] \cup [\pi - \delta_2, \pi)$ the whole unbounded cylinder is the corresponding δ-region (c).

The transformation covered by the most δ-regions is in this case the one (approximately) maximizing the measure of the set $M_\delta(t)$ of point pairs that are at most δ far apart after t is applied. $M_\delta(t)$ is defined exactly as in the case of translations in Equation (3.6).

Approach 3. For the third approach a random sample within one experiment is generated in the following way: Two points a_1, a_2 of the shape A and a point b_1 of B are selected randomly. A point $b_2 \in B$ is selected so that the difference between the y-coordinates of the points b_1, b_2 equals to that of the points a_1, a_2. If there are several points with this property one is chosen randomly. Then there exists a unique shear and translation transformation that maps the point b_1 to a_1 and b_2 to a_2. The corresponding δ-region is defined as a set of transformations that map the point b_1 into the δ-neighborhood of the point a_1 and b_2 into the δ-neighborhood of a_2. That is, it is an intersection of two δ-regions as defined in Approach 2 corresponding to the point pairs a_1, b_1 and a_2, b_2. The parameter δ is the distance tolerance value.

Approach 4. Another possibility is to select two points of each shape randomly, $a_1, a_2 \in A$ and $b_1, b_2 \in B$. Then, a random sample S_A of the shape A consists of a point a_1 and an angle θ_a formed by the line through points a_1, a_2 with the x-axis. Analogously, the sample $S_B = (b_1, \theta_b)$, where θ_b is the angle between the x-axis and the line through points b_1, b_2. The corresponding δ-region is defined exactly as in Approach 1 using two tolerance values δ_1, δ_2, where δ_1 specifies the maximally allowed distance between the points, and δ_2 the difference between the line slopes.

Discussion. Similar to the previously considered transformations, each of the approaches to generating a random sample can provide different matching results in some cases. The most general Approach 2 considers only the distances between the points of the shapes and, therefore, corresponds directly to the intuitive notion of similarity mentioned in the description of the algorithm, Section 3.1. On the negative side, the δ-regions generated by Approach 1 have the largest volume, and therefore, it has the largest search space.

In Approach 1 we use the slope of a tangent line as additional information to reduce the search space for finding the optimal transformation. Since the direction of the tangent line can be unstable, depending on the origin of data, the disadvantages of this method mentioned in the description of Approach 1 for rigid motions, Section 3.3.2, hold also in case of shear transformations.

For Approaches 3 and 4 it can be said that both of them tend to match the general slope information of the shapes and for that shear transformation find the best translation vector. Approach 3 requires a more complex procedure to generate a random sample in each random experiment, but generates δ-regions that comprise transformations that map at least two points of the shape B into the δ-neighborhoods of some two points of the shape A, whereas a δ-region of Approach 4 can match a single point of the shape B to some point of A. Thus, the set of δ-regions produced by Approach 3 is more concentrated and should require less experiments to determine the best matching transformation.

3.3.6 Affine Maps

As described in Section 3.1 a random sample taken from one shape within one random experiment consists of three points. A δ-region in the space of affine transformations corresponding to a pair of random samples from two shapes is then, by definition, the set of transformations that map each point of the sample from shape B into the δ-neighborhood of the corresponding point of the sample from shape A. Such a δ-region is described by equations (3.3), (3.4) or (3.5) for each point correspondence pair of the random samples, depending on the chosen distance metric L_p. Thus, a δ-region is bounded by semi-algebraic surfaces, and for the L_1 and L_∞ metrics it is a convex polytope bounded by six pairs of parallel hyperplanes.

According to our general analysis, an affine transformation which is covered by the most δ-regions maximizes the measure of the set

$$M_\delta(t) = \left\{(a_1, a_2, a_3, b_1, b_2, b_3) \in A^3 \times B^3 | \mathrm{dist}(t(b_i), a_i) \leq \delta, i \in \{1, 2, 3\}\right\}.$$

With a similar observation as for similarity maps, we find that the measure of this set is maximized exactly if the measure of the set of point pairs which are in the δ-neighborhood of each other is maximized. Also by the same argument as for similarity maps we can show that there exists a value $\nu > 0$ such that for any two shapes A and B the maximum of the corresponding probability functions is at least ν:

Lemma 3.7. *Given two shapes A and B modeled by finite sets of n curves with total lengths L_A, L_B respectively and a tolerance value $\delta > 0$ and the class of allowed transformations $\mathcal{T}$ as the group of affine transformations. Let $m = \max(L_A, L_B, n\delta)$. Then $\max_{t\in\mathcal{T}} p_\delta(t) \geq \frac{\delta^3}{m^3}$.*

Proof. The shape B can always be scaled by the factor $\frac{\delta}{D_B}$, where D_B is the diameter of B, so that the diameter of the scaled shape B is δ. If A contains a connected component of length at least δ, then we can place the scaled shape B in such a way that for any point of a part of A of length δ the distance to any point of the scaled B is at most δ. Therefore, the measure of the set $M_\delta(t)$ for that t is at least $L_B^3 \cdot \delta^3$. The corresponding value of p_δ is $p_\delta(t) = \frac{|M_\delta(t)|}{|\Omega|} \geq \frac{L_B^3 \cdot \delta^3}{L_A^3 L_B^3} = \frac{\delta^3}{L_A^3}$. Otherwise, the largest connected component of A must have length at least $\frac{L_A}{n}$. For the transformation t that maps the scaled B to the largest component of A the measure of $M_\delta(t)$ is then at least $\frac{L_A^3}{n^3} L_B^3$ and $p_\delta(t) \geq \frac{1}{n^3}$. Therefore, $\max_{t\in\mathcal{T}} p_\delta(t) \geq \frac{\delta^3}{m^3}$. □

Then, according to Theorem 3.1 the required number of experiments needed to achieve an ε-approximation with probability at least $1 - \eta$ is bounded by $N \in O\left(\frac{m^3}{\varepsilon^2\delta^3} \ln\left(\max\left(\frac{1}{\eta}, \frac{m^3}{\varepsilon^2\delta^3}\right)\right)\right)$.

Matching under affine transformation with the described method suffers under the same shrinking problem as matching under similarity transformations. The shape B can be scaled down by an affine transformation to a shape of diameter at most δ and the scaled shape can be translated to some point of the shape A. Such a transformation has approximately the same value of the probability function p_δ as the transformation that maps the shape B exactly to the shape A if the shapes are congruent.

One possibility to avoid this problem is to discard the sample pairs S_A, S_B if the determinant of the matrix M of the transformation mapping S_B to S_A is below some threshold value.

3.4 Further Ideas and Discussions

In this section we summarize some general observations about the method presented here, and some ideas about further possible variants of the algorithm.

3.4.1 Approximation of the Hit Probability by Grid Counting

As stated in Theorem 3.2 the major time cost of the algorithm is the construction of the arrangement $\mathcal{A}$ of the δ-neighborhoods and the subsequent detection of the deepest cell. This "overhead" can be avoided by applying a grid counting method in order to determine the best matching transformation. In this section we analyse the approximation factor that can be guaranteed by this method and construct an example for which the worst case can actually occur.

Grid counting is the most popular technique used for voting schemes in the pattern recognition community, as for example the case in the articles cited in Section 2.3.1: [1, 31, 48, 55, 57]. Typically, the transformation space is subdivided into equally sized bins, or grid cells. Then for each bin the number of votes that fall into that bin are counted and the bins with the most votes are considered to be the candidates for good matching transformations. A vote is usually a single transformation that maps the sample of one shape exactly to that of the other. The bin is represented by one transformation, for example by one of the corners of the corresponding grid cell.

The obvious advantage of this method, is that adding a vote to a bin can be performed in constant time, and, therefore, the bin containing the maximum number of votes can be determined in time linear in the total number of votes. Given the good time performance of the method, the question is how good are the results.

In the following we describe how this method can be adopted to the votes given by δ-neighborhoods, examine some variants of defining the grid in transformation space and show that counting votes by grid gives a 2^d-approximation of the number of neighborhoods covering the deepest cell in the arrangement $\mathcal{A}$, where d is the dimension of the transformation space. This worst case approximation factor can actually occur in some cases which we will characterize later.

A natural adaptation of the grid counting method to the votes given as regions in the transformation space (which the δ-regions are) is to count a vote for every grid point covered by a region. Of course the grid width should be small enough so that every δ-region covers at least one grid point. Note that in the case of translations in combination with the L_∞-metric and the grid size equal to δ counting votes for the covered grid points is equivalent to counting votes for the grid cells that contain the center of the δ-region.

Translations

We first show that for the axis-aligned grid of width $\gamma \leq \delta$ in the case of translations the grid counting method gives a 4-approximation to the maximum of the hit probability $p_\delta(t)$ (defined in Section 3.2.1) up to a relative error ε. Formally, let t_{opt} denote a transformation maximizing $p_\delta(t)$, t_{g} the grid point covered by the maximum number of the δ-neighborhoods after N experiments, and let $Z(t)$ denote the number of the δ-neighborhoods that contain the transformation t, then $Z(t_{\text{g}})/N \geq \frac{1}{4}(1-\varepsilon)p_\delta(t_{\text{opt}})$.

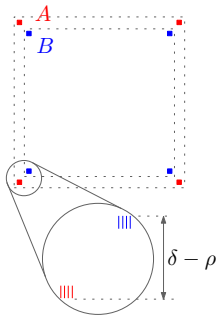
The following example shows that the approximation factor can be as bad as four: Consider shapes A and B that each consist of four small and widely separated parts positioned at the corners of a square. By widely separated we mean that the distance between the parts is significantly larger than the diameter of a part plus 2δ. Further, assume that the parts are equally large, that is, the total lengths of the curves are equal, and the parts of the shape B are equal to the corresponding parts of the shape A. The only difference between A and B is that each part in B is shifted by $\delta - \rho$ towards the center of the shape relative to its position in A, where $\rho > 0$ is

a very small value. In the extreme case the parts are just points in the corners of a square. Since the main focus of this work are shapes modeled by sets of straight line segments, each part of the shapes in the example depicted in Figure 3.18(a) consists of a group of four vertical line segments. Each segment has length three and the distance between segments in one group is one. Thus, each group models a three by three filled square. The distance between two horizontally or vertically aligned groups of the shape A is 300, and the value of δ chosen for this example is 20. Note that although the value of δ is large compared with the size of a single group within a shape, it is feasible with respect to the diameter of the shapes. The value of ρ is chosen to be 2 for visualisation purposes.

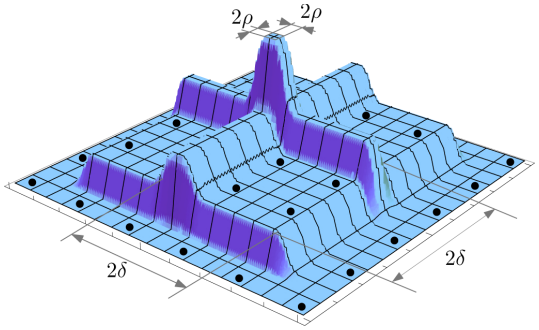
The graph of the hit probability function $p_\delta(t)$ for the shapes A and B in combination with the L_∞ metric is displayed in Figure 3.18(c), and Figure 3.18(b) gives a detailed view of the central group containing the maximum of the function. The support of the hit probability function is subdivided into 9 regions. The four corner regions contain the translations that match exactly one group of the shape B to the diagonally opposite group of the shape A. Note that there is a square in each of these regions with side length $2(\delta - \text{size of a group})$ such that for every translation t in that region every point of the corresponding group in B is mapped into a δ-neighborhood of every point of a diagonally opposite group in A. Therefore, the probability function is constant and has equal values over these square regions, and it decreases to zero along a narrow border around that square. The four regions in the center of each side of the probability function support contain the translation vectors that map one group of the shape B to the opposite group along one side of a square of the shape A, and there is a narrow stripe of width 2ρ that maps two groups on one side of B to two groups on the opposite side of A. Finally, the center region (Figure 3.18(b)) contains the maximum of the probability function. The function $p_\delta(t)$ attains its maximum over a square region with side length 2ρ, which is the set of translations that map each of the four parts of B into a δ-proximity of the corresponding part in A. Around the maximum region there are two stripes with the value $1/2$ of the maximum, which contain the translations mapping exactly two parts of B to the corresponding parts of A. And the larger square regions attached to these stripes contain the translations that map exactly one group of B to the corresponding group of A, thus the value of p_δ over these square regions is $1/4$ of the maximum value.

Note that by shrinking the length of the segments in each group to some arbitrary small value and positioning the segments close together, or just taking one segment in each group, we can get arbitrary close to the extreme case of having just four points in each group, and thus eliminate the regions where the hit probability function decreases gradually

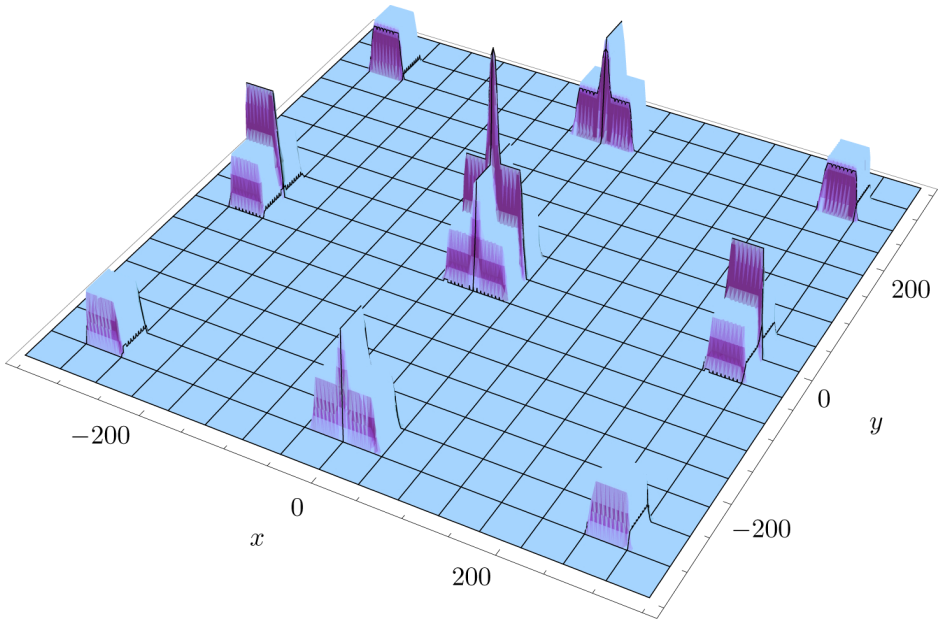
Now if the maximum region happens to lie completely inside one grid cell, the vertices of that cell are positioned in the regions that let only one of the four parts match the corresponding part in A as illustrated in Figure 3.18(b).



(a) Shapes A and B are composed of four groups of line segments. Each shape is enclosed in a dotted square.



(b) A close-up of the center part of the hit probability in translation space. Positions of grid points leading to a 4-approximation ratio are indicated by black disks.



(c) Hit probability function in translation space.

Figure 3.18: An example of shapes and the corresponding hit probability that leads to a 4-approximation by the grid counting method.

According to Lemma 3.1, after a sufficient number of experiments N the estimate $\tilde{p}_\delta(t) = Z(t)/N$ is an ε-approximation of $p_\delta(t)$. In particular it means, that for the maximally covered grid point t_g holds

$$\tilde{p}_\delta(t_\mathrm{g}) \geq (1-\varepsilon)p_\delta(t_\mathrm{g}) \geq (1-\varepsilon)\frac{1}{4}p_\delta(t_\mathrm{opt}) \ .$$

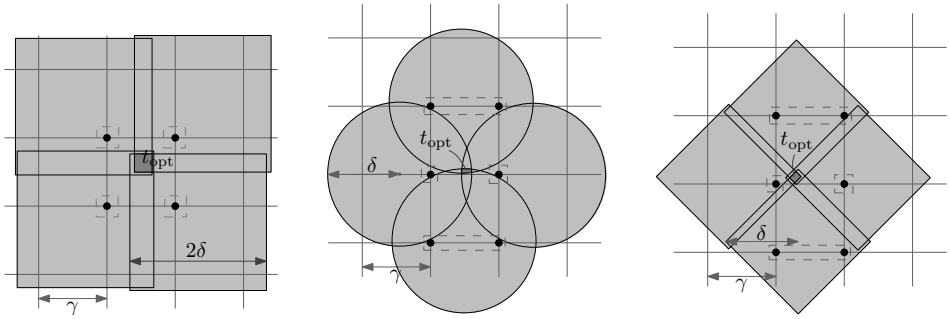
This is also the worst case for the approximation ratio for the described setting. From Lemma 3.1 it follows that $\tilde{p}_\delta(t_{opt})$ is an ε-approximation of the $p_\delta(t_\mathrm{opt})$. Therefore, there are at least $(1-\varepsilon)p_\delta(t_\mathrm{opt})N$ δ-neighborhoods that cover t_opt, which lies insider the grid cell g. Furthermore, due to the choice of the grid width $\gamma \leq \delta$ it holds that if the intersection of a δ-neighborhood with a grid cell is not empty, then either

- the δ-neighborhood covers at least one grid point of the cell, as it is always the case for L_∞ neighborhoods, or
- if the δ-neighborhood does not cover any vertex of the grid cell, then it covers both remaining grid points of the corresponding neighboring cell, which is the case for general L_p metrics, or
- if the δ-neighborhood covers a certain portion of the cell, then it also covers at least two of its vertices.

Then every δ-neighborhood that covers t_opt also covers at least one of the four groups of grid vertices, described in the following, completely, and therefore, at least one of these groups is covered by at least $1/4$ of the δ-neighborhoods. The groups can be the four grid vertices surrounding t_opt, Fig. 3.19(a), which is always the case for L_∞ metric. For other L_p metrics it can also be the case that the votes can be split between two vertices of a cell connected by an edge and two groups of two vertices, that belong to two edges parallel to the first one as depicted in Figure 3.19(b) for L_2 δ-neighborhoods and 3.19(c) for L_1.

Note that for grid size $\delta \leq \gamma \leq 2\delta$ for general L_p metrics no 4-approximation guarantee can be given. The example in Figure 3.20 shows a possible distribution of the votes covering the maximum region in translation space, that are split into six groups of which every group contains different grid points. Thus, each of the grid points gets only $1/6$ of the votes that contain the deepest cell in the δ-neighborhood arrangement.

Reducing the grid size. It is rather easy to see that reducing the distance between the grid points does not help to improve the worst case approximation factor. Intuitively, the probability of getting a bad approximation, or a bad split of votes should be smaller the smaller we choose the grid size γ. In fact, the probability that the maximum region lies completely inside one grid cell, and thus the δ-neighborhoods covering it can be split over the four grid points, is proportional to $(\gamma-\rho)^2/\gamma^2$,



(a) L_∞ (b) L_2 (c) L_1

Figure 3.19: Worst case distributions of votes in case of translations in combination with different L_p metrics.

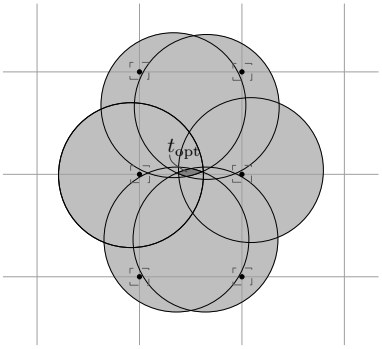


Figure 3.20: An example of votes distribution that gives in case of grid width $\delta \leq \gamma \leq 2\delta$ for L_2 δ-neighborhoods a 6-approximation of the maximum of $p_\delta(t)$.

where 2ρ is the side length of the maximum region. However, in our case the underlying hit probability function $p_\delta(t)$ is not a continuous function, and the size of the maximum region 2ρ can be arbitrarily small. Therefore, the probability of getting a 4-approximation can be arbitrarily large.

Shifted grids. Similarly, using an additional grid G' of width γ which is shifted by $\gamma/2$ relatively to the original grid G, as shown in Figure 3.21, does not improve the worst case approximation factor. By the same argument as above we can say

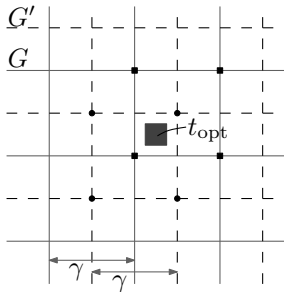


Figure 3.21: Grid counting with two shifted grids.

that the probability of the maximum region lying completely inside the intersection of two grid cells g of G and g' of G' can be arbitrarily large, in which case the vertices of the cells in both grids achieve only a four approximation.

Rigid Motions

In the case of rigid motions for the grid counting method the transformation space is usually parameterized by the rotation angle and translation vector instead of the four-dimensional parameterization by the parameters of the rotation matrix and translation vector. Due to the heterogeneity of the transformation parameters it is reasonable to use a nonuniform grid, that is, to choose a different grid width for the rotation angle dimension than for the translation vector dimensions.

Recall that in Approach 2 for rigid motions we already had two values to express the proximity of shape samples: δ_1 for the maximal distance between two points that are considered close, and δ_2 for the maximal allowed difference in the directions of the tangent lines. These values give us some obvious restrictions for the grid widths: The grid width in translation plane γ_1 should be $\gamma_1 \leq \delta$ for the reasons explained above, and the grid width in direction of the rotation angle γ_2 should be chosen as $\gamma_2 \leq \delta_2$.

For the Approach 1 it is no longer clear how dense the grid should be for the rotation angle. One possibility is to define some additional tolerance value comparable to the δ_2 in Approach 2, which denotes how fine we want to determine the rotation factor. What we additionally want to achieve is a condition similar to that

for the translation case, which assures that if a δ-region intersects a grid cell, then it either contains at least one of the vertices of that grid cell, or it contains at least two vertices of the corresponding neighboring cell.

The details about the δ-neighborhoods in the space of rigid motions are given in the Section 3.3.2, for this section we only give a brief description. A δ-region corresponding to a sample pair $S_A = \{a\}, S_B = \{b\}$ is a spiral tube of height 2π with a base corresponding to a δ-neighborhood in the image space. An example of such δ-neighborhood for L_2 distance measure is depicted in Figure 3.5. The projection of the tube to the translation plane is an annulus centered at a of radius equal to the distance of the point b to the origin and of width δ.

We are now interested in the minimal stretch in the direction of the rotation angle of a δ-neighborhood. Obviously, there are points on the boundary of the δ-neighborhood with stretch zero, therefore, we consider the minimal stretch σ of the tube at the points with offset at least γ_1 from the boundary in the cross-section parallel to translation plane. Then if the grid width in direction of the rotation angle is $\gamma_2 \leq \sigma/2$, we can be sure that a δ-neighborhood can not "go through" the grid cells, that is, have a non-empty intersection with more than one consecutive grid cells without covering any vertices of these cells.

Consider a disc $\mathcal{B}_1$ of radius $\delta - \gamma_1$ with respect to a chosen distance metric L_p with center b_1 on a circle $\mathcal{C} = (c, r)$, and a disc $\mathcal{B}_2$ with respect to the same metric L_p of radius δ with the center b_2 on the same circle $\mathcal{C}$ as depicted in Figure 3.22. Then for a fixed position of b_1 on the circle $\mathcal{C}$ the stretch σ' at that position is the maximum angle formed by b_1 and b_2 in the circle $\mathcal{C}$ for which the disk $\mathcal{B}_2$ covers completely the disk $\mathcal{B}_1$. The stretch σ is the minimum σ' over all positions of b_1.

For the Euclidean distance metric L_2, if two disks are tangent and one of them is inside of the other, then the Euclidean distance between their centers is exactly the difference of the radii. Thus, the maximum allowed Euclidean distance between the points b_1 and b_2 is γ_1 for every position of the point b_1. By the law of cosines the corresponding angle σ is

$$\sigma = \arccos\left(1 - \frac{\gamma_1^2}{2r^2}\right) \quad . \tag{3.14}$$

For general L_p distance metrics, with $p \geq 2$, the stretch at any position of b_1 is at least σ, where σ is defined by Equation 3.14. Let q be a point in $\mathcal{B}_1$, then $d_p(q, b_1) \leq \delta - \gamma_1$. Due to the fact that for all points $q_1, q_2 \in \mathbb{R}^2$ $d_p(q_1, q_2) \leq d_2(q_1, q_2)$ and from the triangle inequality, it follows that the distance between q and b_2 is at most δ:

$$d_p(q, b_2) \leq d_p(q, b_1) + d_p(b_1, b_2) \leq \delta - \gamma_1 + d_2(b_1, b_2) = \delta - \gamma_1 + \gamma_1 = \delta \ ,$$

where d_p denotes the distance with respect to L_p. Furthermore, it is easy to see that for positions of b_1 corresponding to the angles $\left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\} - \frac{\sigma}{2}$ in $\mathcal{C}$ the stretch is exactly σ.

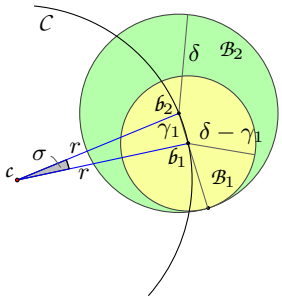


Figure 3.22: Disks $\mathcal{B}_1$ and $\mathcal{B}_2$ with respect to the L_2 metric in extreme position, i.e., $\mathcal{B}_1$ is contained in $\mathcal{B}_2$ and the distance between their centers is maximized.

In case of L_1 distance metric the minimum Euclidean distance between the centers of the disks $\mathcal{B}_1, \mathcal{B}_2$, such that points on boundary of the disks coincide, is $\gamma_1/\sqrt{2}$: A disk of radius δ with respect to the L_1 distance is a square with diagonal length 2δ with axis-parallel diagonals. Two equally oriented nested squares have a common boundary point iff they have at least one collinear side. Thus, we are interested in the minimal distance from b_1 to b_2 if $\mathcal{B}_1$ and $\mathcal{B}_2$ have collinear sides. For the sake of simplicity, let the center of $\mathcal{B}_2$ be placed at the origin, and the collinear sides be the sides in the positive coordinate quadrant, as depicted in Figure 3.23. Then for every point $q = (x, y)$ on the boundary of $\mathcal{B}_1$ and $\mathcal{B}_2$ holds: $x \geq b_{1x} \geq b_{2x} = 0$ and $y \geq b_{1y} \geq b_{2y} = 0$. Further, since q is a point on the boundary of $\mathcal{B}_2$ and $\mathcal{B}_1$:

$$x + y = \delta \qquad \text{and}$$
$$x - b_{1x} + y - b_{1y} = \delta - \gamma_1 \ .$$

From these equations follows $b_{1y} = \gamma_1 - b_{1x}$. Plugging in this values into the expression for the Euclidean distance we get $d_2^2(b_1, b_2) = 2b_{1x}^2 - 2\gamma_1 b_{1x} + \gamma_1^2$. This expression is minimized for $b_{1x} = b_{1y} = \gamma_1/2$. Thus, the minimal Euclidean distance between the centers is $\gamma_1/\sqrt{2}$.

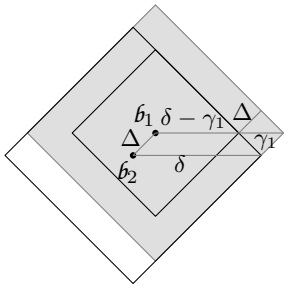


Figure 3.23: Minimal Euclidean distance Δ between centers of L_1-disks, such that points on the boundary of the disks coincide.

The minimal stretch of the δ-neighborhood tube in case of L_1 metric is then

$$\sigma = \arccos\left(1 - \frac{\gamma_1^2}{4r^2}\right) \quad . \tag{3.15}$$

Then by selecting the grid width in direction of rotation angle as $\gamma_2 \leq \sigma$, where σ is given by Equation (3.14) or (3.15) depending on the underlying distance metric, we can guarantee in case of rigid motions the worst case approximation factor 8 for the maximum value of $p_\delta(t)$. For Approach 1 it is, of course, meaningful to choose $\gamma_2 \leq \min(\delta_2, \sigma)$.

Higher Dimensional Transformations

In general, for a grid in d-dimensional transformation space the approximation factor of the grid counting method can be as bad as 2^d, since the maximum region can be located completely in the interior of one grid cell and each of the δ-neighborhoods that cover the maximum region might contain only one of the vertices of that cell. But this factor can no longer be guaranteed as the worst case approximation factor. As we have seen above, the δ-neighborhoods behave "nicely" in the case of translations, in a sense that all δ-neighborhoods have equal shape and size and that the δ-neighborhoods that cover the maximum region of the $p_\delta(t)$ probability function, and therefore contribute to the maximally covered cell of the arrangement, also cover the neighboring grid points. For higher dimensional transformations it can not be guaranteed in general, since the δ-neighborhoods are no longer uniformly sized and are no longer convex.

Thus, the grid counting is a purely heuristic method for the higher dimensional transformations.

Discussion

The bounds on the approximation factor for the grid counting method presented in this section are rather of negative kind, since the approximation factor of 2^d, where d is the dimension of the transformation space, does not seem to be suitable for practical usage. Nevertheless, the method is widely used in the computer vision and pattern recognition community.

The obvious advantage of the grid counting method over, for example, the approximation by arrangement is the running time, which is linear in the number of samples. Additionally, grid counting is reported to perform well in practice. In fact, the example that we used here to show that the worst case approximation factor can actually occur has a special structure which is unlikely to occur in practical applications. Observe that if the size of the components or the relation between the tolerance value δ and the distance between the components of the two shapes in our example are modified slightly, then the approximation factor improves. Thus we can expect the grid counting method to perform significantly better in practice than the worst case approximation factor. In fact, experimental comparison of the

grid counting method and the arrangement method for the class of translations in combination with the MPEG-7 Shape B dataset and with the trademark images showed only small difference in matching results. We actually used the grid counting method in many experiments because of its better runtime performance.

3.4.2 Properties of the Probabilistic Algorithm

Robustness

In many applications the geometric shape data is gained through some digitalization process from analog data. Examples of such digitalization processes are pictures of a real world scenes taken with a digital photo camera, scans of graphical images from paper originals, or data gathered from some special purpose devices, such as range data scanner. After such a digitalization process we still do not have geometric data, but rather a pixel image or a set of points without any connectivity information. The next step is typically a vectorization process, which might include edge detection, grouping, or curve reconstruction. During both steps toward geometric data, digitalization and vectorization, some errors might occur, e.g., parts of important data might be lost, or some additional data might be added. In many cases it is difficult to automatically distinguish between the important and unimportant data. Therefore, if we need to compare shapes gained from a possibly noisy source, it is important to know whether the matching algorithm we use is robust against possible disturbances in the shape data.

In this section we discuss robustness of the probabilistic method against four types of disturbance as described by Hagedoorn [33] in his PhD thesis. There he distinguishes between noise, crack, deformation, and blur, which are illustrated in Figure 3.24.

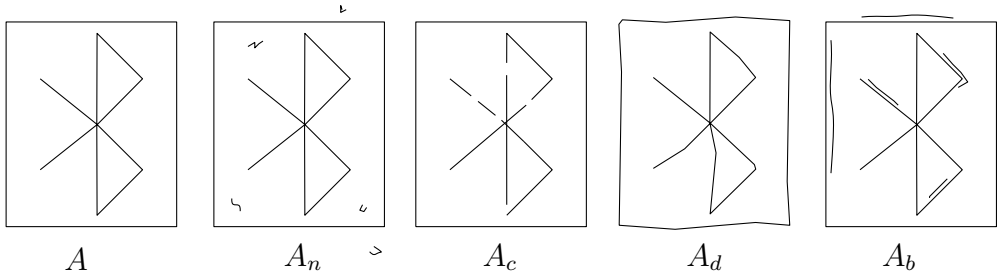


Figure 3.24: Types of disturbances: A – undisturbed shape, A_n – shape A with noise added, A_c – shape A with cracks, A_d – deformed shape A, A_b – shape A with blur added.

Informally, a similarity measure f is said to be robust against a certain type of disturbance if for three shapes A, A' and B, where A' is the shape A with small amount of disturbance added, the difference between the values of $f(A, B)$ and $f(A', B)$ is small. Recall that in Section 3.2.2 we discussed two similarity mea-

sures associated with the algorithm: the measure of the set M_δ and the value of the hit probability p_δ. In the following we argue that the probabilistic method is robust against noise, crack, and deformation, and to some extent against blur for the resemblance measures $\mu_\delta(A,B) = \max_{t\in\mathcal{T}}\{(a,b)\in A\times B \mid \operatorname{dist}(a,t(b))\le\delta\}$ and $\rho_\delta(A,B) = \frac{\mu_\delta(A,B)}{|A\times B|}$, where $\mathcal{T}$ is the class of allowed transformations.

Noise. Noise or "outliers" are typically small isolated features of a shape. In our setting we could call curves of small length, for which the distance to the remaining parts of the shape is large, noise.

Let ε be the measure of noise added in A', that is $|A'\setminus A| = |A'|-|A| = \varepsilon$. Then, since we only added some curves to A and did not remove any, the measure of the set $M_\delta(t)$ for the shapes A' and B is at least as large as the measure of $M_\delta(t)$ for the shapes A and B, that is $\mu(A',B)\ge\mu(A,B)$. Furthermore, $\mu(A',B)$ differs from $\mu(A,B)$ by at most the amount of added noise matched to some parts of B, which is bounded by $\varepsilon\,|B|$. Thus, the difference between the resemblance measure of shapes A and B and resemblance measure of shapes A' and B is directly proportional to the amount of noise added to A.

Similarly, for measure ρ it holds:

$$\rho(A',B) \ge \frac{\mu(A,B)}{|A\times B|+\varepsilon\,|B|} = \rho(A,B)\frac{|A|}{|A|+\varepsilon}$$

and

$$\rho(A',B) \le \frac{\mu(A,B)+\varepsilon\,|B|}{|A\times B|+\varepsilon\,|B|} = \rho(A,B)\frac{|A|}{|A|+\varepsilon} + \frac{\varepsilon}{|A|+\varepsilon}\ .$$

So the difference between $\rho(A,B)$ and $\rho(A',B)$ is small for small ε.

Crack. A shape A' is said to have cracks if there are small gaps in A' which are not present in the original shape A. So a single curve of A might be subdivided in multiple curve segments in A' which have small distance between their endpoints.

Let ε be the total measure of curve segments of A that are missing in A', i.e., $|A\setminus A'| = |A|-|A'| = \varepsilon$. Then analogously to the noise disturbance we can derive relations of the resemblance measures μ and ρ for shape pairs A,B and A',B:

$$\mu(A,B)-\varepsilon\,|B| \le \mu(A',B) \le \mu(A,B)$$
$$\rho(A,B)\frac{|A|}{|A|-\varepsilon} - \frac{\varepsilon}{|A|-\varepsilon} \le \rho(A',B) \le \rho(A,B)\frac{|A|}{|A|-\varepsilon}\ .$$

Again, we can conclude that for small total amount of gaps the difference between resemblance measures for shape pairs A,B and A',B is also small.

Deformation. If some parts of a shape A are displaced by a small distance from their original position the resulting shape A' is said to have deformation disturbance.

Observe, that it does not necessarily mean that a complete curve contained in A is transformed by a nearly identity transformation, but rather some segments of the curves in A may be independently transformed. So for example, a single straight line segment in A might be represented by a polygonal cure in A'.

The difference of the resemblance measures for the pairs of shapes A, B and A', B depends on the measure of deformed parts of A as well as on the amount of displacement. As in the previous cases it still holds that the deformation of small parts of A results in a small difference between the corresponding resemblance measures. Additionally, in many cases deformation of large parts of A with small displacement results in a small change of resemblance measures, as long as matching parts of A and B can still be mapped into a δ-neighborhood of each other.

Blur. A shape A' is said to have blur if some parts of the contours of the original shape A are doubled with a small offset in A'. The duplicates of contour parts might be additionally deformed. As in the case of deformation there are two quantities in which blur can be measured: the total size of the contour duplicates, and the offset between the duplicates and the original curves.

The probabilistic method and the underlying resemblance measures are robust against blur to only a limited extent. If blur is evenly distributed along the curves of the shape A with small offset, then the best matching transformation found by the algorithm for the blurred shape A' and some shape B is close to the transformation found for the shapes A and B. The difference of the resemblance measure μ for the pairs A, B and A', B is directly proportional to the size of the doubled contours of the matched parts of A. The size of the doubled contours has less influence on the measure ρ since, assuming evenly distributed blur, the ratio of the size of blurred matched contours to the total size of the blurred shapes is close to the ratio of the size of matched contours to the total size for the original shapes.

However, large amount of blur even with small offset that affects only few parts of a shape might significantly influence the matching results with regard to the best matching transformation as well as the value of resemblance measures.

Complete and Partial Matching

In this section we discuss the applicability of the probabilistic method to the problem of complete and partial matching. In general, the problem of *complete-complete matching* is defined as rating the similarity between the complete shape A and the complete shape B. If we ask how good does the complete shape A match some part of the shape B we have a *complete-partial matching* problem. And if we want to know how similar are some parts of A to some parts of B we ask for a *partial-partial matching*. Observe that the problem of partial-partial matching is not well-defined since arbitrary small parts of two shapes can always be matched arbitrarily well to each other. Therefore, there should be a certain trade-off between the quality of match and the size of matched parts.

As mentioned in Section 3.2.2 the resemblance measure underlying the matching algorithm described here is the maximum measure of the set M_δ, which is the set of sample pairs that can be mapped into a δ-proximity of each other. We also argued in Section 3.3 that, for the samples consisting only of points of the shapes, it is equivalent to maximizing the measure of the set of point pairs that are δ-close to each other. The hit-probability p_δ is the measure of the set M_δ normalized by the measure of all possible sample pairs. So also p_δ indicates the ratio of the matched parts of two shapes. Thus, the probabilistic algorithm performs a partial-partial matching, where the trade-off between the quality of match and the size of the matched parts is resolved by letting the user specify the desired quality of match. The algorithm then determines how large the matching parts can be, but not which parts are matched. If the information about matched parts is required and additional verification step should be performed.

The probabilistic method can also be used to perform complete-partial matching, that is, to evaluate the resemblance of the complete shape A to some part of the shape B. For this purpose we can use the measure of the set M_δ normalized by the size of the shape A. This variant was also discussed in the Section 3.2.2. Finally, the maximum of the normalized resemblance measures for each of the shapes can be used to determine complete-complete similarity.

We also thought about the problem of determining the optimal tolerance value δ for two given shapes automatically. The main idea is the following: Let μ denote the maximum measure of the set $M_\delta(t)$ over all allowed transformation for shapes A and B. Consider μ as a function of the tolerance value δ. Obviously, $\mu(\delta) = \max_{t\in\mathcal{T}} M_\delta(t)$ is monotone increasing.

Let us consider the behavior of the function $\mu(\delta)$ in a few settings: First let the shape B be a possibly shifted and rotated copy of the shape A. Then already for a small δ the complete shapes A and B can be perfectly matched. With increasing δ the growth of μ should be nearly linear, since every point of one shape can be matched against a segment of the other shape with length approximately δ. So we do not expect the function μ to have large changes in growth over a large interval of δ values. Further, even for small δ the value of μ should be relatively large.

If the shapes A and B are very different, then for small δ the value of μ should also be very small. Further, with increasing δ only small parts of shapes will be matched against each other. Therefore, also in this case we do not expect the function μ to have significant changes in growth, until for some large value of δ large parts of the shapes can be mapped into δ-proximity of each other.

Now assume that the shapes A and B have similar parts. Further, let δ' be the smallest value of δ for which large parts of A and B can be mapped into δ-proximity of each other. Then we would expect that the function μ grows fast in some interval before δ', and slow after. If there are several parts in A and B that can be matched within different tolerance values, then there should be correspondingly many values of δ where the growth behaviour of μ changes from fast to slow. An example of such

$\mu(\delta)$ function is depicted in Figure 3.25, the circles mark the possibly interesting changes in the graph of the function.

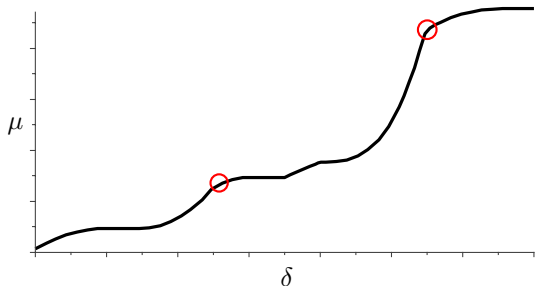


Figure 3.25: An example of the measure $\mu(A, B)$ as a function of δ. Circles mark the possibly interesting spots in the graph of μ.

When the values of δ for which large parts of the shapes match and the size of the matching parts for each of the values are determined, some trade-off evaluation function is needed to determine the best precision-size combination.

The described approach evokes several open questions, starting with computation or approximation of the function $\mu(\delta)$ in reasonable time. Observe that the method presented in this work approximates the value of μ for one value of δ. Further, it would be interesting to examine whether the behavior of the $\mu(\delta)$ function actually corresponds to the one conjectured here for the practice-relevant shapes. It is not immediately clear how the δ values of interest can be formally characterized and automatically detected, and whether the relevant values can be found without computing the complete $\mu(\delta)$ function.

Alternative Distributions

The probabilistic approach presented here can be combined with alternative predefined mass distribution functions along the curves of the shapes. In the description and analysis of the method above we suggested the generation of random samples under uniform distribution with respect to the length of curves. This implies that every part of the shape of equal length is equally important and the best matching transformation is the one that maps the parts of maximal length into proximity of each other, which is a reasonable definition for many applications.

However, if it is known a priori that some parts of the shapes are more important than the others, then by assigning those parts higher probability we can achieve that the transformations mapping important parts into proximity of each other receive higher weight.

Trademark images are a good example of shapes containing components of different importance. Many trademark images are enclosed in frames, which can be a rectangle, circle or an oval shape. Such a frame curve usually has a relatively large length and, therefore, a relatively high weight within the shape. However, it is

clear that for the human perception the symbol within the frame has much higher importance. So we would like to have shapes with the same trademark symbol enclosed in differently shaped frames to be rated more similar than two shapes with circular frames but different trademark symbols. This can be achieved by assigning lower probability to the frame curves and higher probability to the main symbol, thus changing the mass distribution within the shape. If the frame detection, or detection of other unimportant features is performed automatically and the parts of the shape can be classified with some certainty factor, the reduction of mass can also be performed proportionally to that factor.

This idea can be generalized to arbitrary mass distributions along the curves instead of assigning importance to some parts of the shapes. For natural contour shapes Attneave [10] suggested in 1954 that information along visual contours is concentrated in regions of high magnitude of curvature, rather than being distributed uniformly along the contour. This suggestion was confirmed by several subsequent experimental studies. Furthermore, Feldman and Singh [25] give a formal derivation of this claim, yielding a mathematical expression for information concentration along the contour as a function of contour curvature. Moreover, they extend Attneave's claim to incorporate the role of sign of curvature, not just magnitude of curvature. Thus, for matching contours of natural objects the mass distribution along the curves could be adjusted according to the information concentration formulas derived by perception psychologists.

Observe that the general statements of Section 3.2 about the underlying similarity measure for the probabilistic algorithm remain true for alternative mass distributions. Also the interpretation that the best matching transformation for two shapes is the one that maps the largest parts into δ-proximity of each other is still true. Although the largest parts are no longer the parts of the maximum possible length, but the parts with maximum total weight, or importance, that can be matched under allowed transformations.

3.4.3 Higher Dimensions

The probabilistic method described in this thesis is not restricted to matching shapes represented by planar curves. Rather there are several possibilities to extend the results presented here to matching shapes in higher dimensions. Observe that there are two types of dimensions that can be considered: the dimension of shapes, which is one in this work, and the dimension of the space containing shapes, which we call image space. The dimension of the image space is two in this work.

The dimension of the transformation space depends on the dimension of the image space and the class of transformations. So for example, the space of translations of the d-dimensional image space has also the dimension d. A rotation in a d-dimensional space has $d(d-1)/2$ degrees of freedom. Thus, rigid motions in d-dimensional space have $d(d+1)/2$ parameters. Similarity transformations have one additional parameter – scaling, i.e., the space of similarity transformations of d-

dimensional image space has dimension $d(d+1)/2+1$. The more general affine transformations have $d \times d$ parameters of the linear transformation matrix plus d parameters for the translation vector.

We expect that properties and approximation bounds similar to those given in this work can be shown for the probabilistic method applied to matching of shapes of dimension at most $d-1$ in d-dimensional image space with an appropriate choice of random samples. Mohajer [46] studied in her Master's thesis experimentally the probabilistic method for matching surfaces, that is, two-dimensional objects, in three-dimensional space under translations, rigid motions and similarity transformations.

The situation is different if a similar probabilistic approach is applied to matching shapes of dimension d in d-dimensional image space. Together with Helmut Alt and Daria Schymura we showed in [7] that for polygonal shapes in the plane and transformation classes translations and rigid motions the probabilistic algorithm finds a transformation approximately maximizing the area of overlap of two given shapes. The analysis of the properties of the probabilistic method in combination with other transformation classes is a subject of ongoing research by Daria Schymura. The proof that the underlying similarity measure in case of translations is the area of overlap can be adopted to higher dimensions in a straight forward manner, whereby the similarity measure is the d-dimensional volume of the intersection of two shapes.

The generalizations of the algorithm to higher dimensions are rather of theoretical interest, since for practical use the running time of the clustering by arrangement of δ-regions becomes infeasible. For practical purposes it is reasonable to modify our technique and enhance it with heuristic methods.

Of course, the probabilistic method presented here is also applicable for matching discrete point sets. In discrete setting the measure of a point set is to be understood as the number of points contained in that set. It is then closely related to a randomized version of the generalized Hough transform. The underlying resemblance measure is then the maximum possible number of correspondences between the points of two shapes. Due to consideration of δ-regions in transformation space instead of single transformation vectors, two points $a \in A$ and $b \in B$ are considered to be matched not only if b is mapped exactly to a but also if it is mapped into a δ-neighborhood of a for some specified tolerance parameter δ.

3.4.4 The Adaptive Heuristic

In this section we briefly describe an idea of the adaptive clustering heuristic. The motivation for this heuristic is the observation that the probabilistic method generates many sample pairs that do not result in a good correspondence between the shapes, and thus produce δ-regions in sparsely covered regions of the transformation space. Another simple observation we can make is that if two shapes have large matching parts for some tolerance value δ_1 then these parts still match for a larger tolerance value $\delta_2 > \delta_1$.

The main idea of the adaptive heuristic is to set the tolerance parameter δ to

some large value at the beginning of the algorithm. This initial value should be selected significantly larger than that specified by the user. After some (not too large) number of experiments the clustering is performed and densely sampled regions in the transformation space are identified.

In the next stage the value of δ is reduced and the generation of samples is restricted to only such pairs that produce a δ-region in a dense region of the transformation space. After another set of random experiments clustering is performed again and the region of allowed transformations is further restricted. This process is repeated until δ is reduced to the user specified value.

Since the adaptive matching process changes the distribution of votes compared to the general probabilistic matching, the measure of resemblance between the shapes can no longer be derived directly from clustering results. Therefore, an additional verification procedure for the best candidate transformations might be necessary.

Alternatively to the predefined tolerance value δ, we could let the user specify the required size of matching parts of the shapes and iteratively find the smallest value of δ for which the matched parts are still large enough. In this scenario the minimum value of δ for which the shapes still match to a sufficiently large extent serves as the dissimilarity measure and no additional verification procedure is needed.

The major open problem in this heuristic is the generation of sample pairs in such a way that the corresponding δ-region is contained in a certain region of the transformation space.

Chapter 4

Analysis of the Density Function of the Probability Distribution

In this section we analyze the probability distribution of the "votes" in the transformation space. That is, if we record for each sample pair the transformation that maps the sample of the shape B to the sample of the shape A, we get a certain distribution of transformation vectors. For some of the transformations considered in this work (translations and rigid motions) the density function induced by the random experiment can be analysed and a closed form of that function is given here. For other transformations, such as similarity maps, an exact analysis seems to be very hard. Even deriving a closed form for the density function of the distribution of the scaling factor for rather simple shapes, e.g., shapes consisting of two straight line segments, involves an extensive case distinction. Therefore, we restrict the analysis of the scaling factor distribution to just few simple shape classes and report the experimental density function evaluation on the MPEG-7 dataset in Appendix A.

For the case of translations and for the rigid motions we show that the density function of the induced probability distribution corresponds to the generalized Radon Transform. A general form of the Radon Transform with respect to images as defined in [30] is given in Section 2.3.3.

In our case the shapes are one-dimensional objects in two-dimensional image space. Let $C_A : \mathbb{R}^2 \to \mathbb{R}$ denote the constraint function of the shape A, that means $C_A(p) = 0$ iff a point $p \in A$. And let $C_B : \mathbb{R}^2 \times T \to \mathbb{R}$ be the constraint function of the shape B with transformation $t \in T$ applied, $C_B(p,t) = 0$ iff $p \in t(B)$ or, equivalently, iff $t^{-1}(p) \in B$, where t^{-1} denotes the inverse transformation. Assume that the constraint functions fulfill the property that a 2-dimensional integral of $\delta(C_A(p))$, or $\delta(C_B(p,t))$ for a fixed t respectively, over a region $R \subset \mathbb{R}^2$ evaluates to the 1-dimensional volume (length) of the part of the shape contained in R. Then the Equation (2.4) defining the Radon transform of the shape A with respect to the

shape B can be written as:

$$\mathcal{R}_B(A,t) = \int_{\mathbb{R}^2} \delta(C_A(p))\delta(C_B(p,t))\,\mathrm{d}\,p \ . \tag{4.1}$$

The Radon transform provides a mapping from image space to parameter space, which is the transformation space here. The function created in the transformation space contains peaks for those t, for which $t(B)$ is (partially) present in image A. For other transformations t' the function evaluates to a finite number proportional to the number of intersections between $t'(B)$ and A.

Let X_A and X_B denote two-dimensional random variables that contain coordinates of the points drawn randomly, uniformly distributed with respect to length from the shapes A and B, respectively. The probability density functions $f_A(p)$ of X_A and $f_B(p)$ X_B are defined as

$$f_A(p) = \frac{1}{|A|}\delta(C_A(p)) \quad \text{and} \quad f_B(p) = \frac{1}{|B|}\delta(C_B(p,0)).$$

f_A and f_B fulfill the properties of probability density functions: $\int_{\mathbb{R}^2} f_A(p)\,\mathrm{d}\,p = 1$ and $\Pr(X_A \in R \subset \mathbb{R}^2) = \int_R f_A(p)\,\mathrm{d}\,p = \frac{|R \cap A|}{|A|}$, analogous holds for f_B.

4.1 Translations

Let X_T denote the two-dimensional random variable $X_T = X_A - X_B$, that is, X_T contains the coordinates of the translation vector generated by one random experiment of the probabilistic algorithm. The random variable X_T is a sum of two independent random variables X_A and $-X_B$ and, thus, has the following density function, see [32]:

$$\begin{aligned} f_T(t) &= \int_{\mathbb{R}^2} f_A(a) f_B(a-t)\,\mathrm{d}\,a \\ &= \frac{1}{|A|\,|B|}\int_{\mathbb{R}^2} \delta(C_A(a))\delta(C_B(a-t,0))\,\mathrm{d}\,a \\ &= \frac{1}{|A|\,|B|}\int_{\mathbb{R}^2} \delta(C_A(a))\delta(C_B(a,t))\,\mathrm{d}\,a \\ &= \frac{1}{|A|\,|B|}\mathcal{R}_B(A,t) \qquad \text{by equation (4.1)} \end{aligned}$$

Therefore, the density function of the probability distribution in the translation space corresponds to a generalized Radon transform weighted by a factor $\frac{1}{|A||B|}$.

In the following we deduce a closed formula for the Radon transform for the case where shapes A and B are represented by finite sets of straight line segments.

The generalized Radon transform of straight line segments. Let s_a and s_b be straight line segments defined by their end points a_1, a_2 and b_1, b_2. Consider the

straight lines g_a and g_b supporting these segments. The normal form, also called Hesse standard form, of a line g_a is

$$g_a : k_a - x\cos\theta_a - y\sin\theta_a = 0 \ ,$$

where θ_a is the angle of inclination of the normal and k_a is the length of the normal. The values of $\cos\theta_a, \sin\theta_a$ and k_a can be found as $\cos\theta_a = (a_{1y} - a_{2y})/|s_a|$, $\sin\theta_a = (a_{2x} - a_{1x})/|s_a|$ and $k_a = (a_{1y}a_{2x} - a_{1x}a_{2y})/|s_a|$. The constraint function for g_a is $C_A(x,y) = k_a - x\cos\theta_a - y\sin\theta_a$ and it fulfills the integral constraints required above. The parametric form of g_a can then be defined as

$$\begin{aligned} g_{ax}(s) &= k_a\cos\theta_a - s\sin\theta_a \\ g_{ay}(s) &= k_a\sin\theta_a + s\cos\theta_a \end{aligned} \tag{4.2}$$

for $s \in \mathbb{R}$. The segment s_a corresponds then to $s \in I_a \subset \mathbb{R}$ for some interval I_a. Analogously we can find normal and parametric form for g_b.

The Radon transform of the line g_a with respect to the segment s_b for transformation parameter $t = (t_x, t_y)$ can be written according to the form (2.3) in Section 2.3.3 in a following way (see also [57]):

$$\begin{aligned} \mathcal{R}_{s_b}(g_a, t) &= \int_{s\in I_b} \delta\left(k_a - (g_{bx}(s) + t_x)\cos\theta_a - (g_{by}(s) + t_y)\sin\theta_a\right) \mathrm{d}\, s \\ &= \int_{s\in I_b} \delta\left(k^* - k_b\cos(\theta_b - \theta_a) + s\sin(\theta_b - \theta_a)\right) \mathrm{d}\, s \ , \end{aligned}$$

where $k^* = k_a - t_x\cos\theta_a - t_y\sin\theta_a$.

Now if the line segments are not parallel, that is $\sin(\theta_b - \theta_a) \neq 0$, then for every translation t there is at most one value of s such that $k^* - k_b\cos(\theta_b - \theta_a) + s\sin(\theta_b - \theta_a) = 0$. Furthermore, for $t \in g_a \oplus (-s_b) = T_b'$ there exists exactly one value of s and for all other t none. Thus, for $t \in \mathbb{R}^2 \setminus T_b'$ the Radon transform $\mathcal{R}_{s_b}(g_a, t)$ is zero, and for $t \in T_b'$ there is exactly one $s \in I_b$ where the argument of the delta function evaluates to zero, and according to the integration rules of the delta function

$$\begin{aligned} \mathcal{R}_{s_b}(g_a, t) &= \frac{1}{|\sin(\theta_b - \theta_a)|} \int_{s\in I_b} \delta\left(\frac{k_a - k_b\cos(\theta_b - \theta_a)}{\sin(\theta_b - \theta_a)} + s\right) \mathrm{d}\, s \\ &= \frac{1}{|\sin(\theta_b - \theta_a)|} \end{aligned}$$

If, instead of the line g_a, we consider the segment s_a the region in the translation space where the Radon transform is not zero reduces to $T_b = s_a \oplus (-s_b)$, within this region, though, the constraint function for the segment s_a is exactly the same as for the line g_a and, thus, the value of the Radon transform stays unchanged. Figure 4.1 illustrates the probability density function for two non-parallel segments.

If the segments are parallel we have a one dimensional translation case. As pointed out in [57] a function $g : \mathbb{R}^2 \to \mathbb{R}$ can be represented as $\int_{\mathbb{R}^2} g(x^*, y^*)\delta(x - x^*)\delta(y - y^*)\,\mathrm{d}\, x^*\,\mathrm{d}\, y^*$. Using parameterization of the segment we can represent s_a

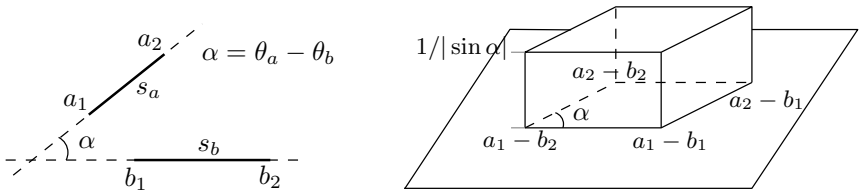


Figure 4.1: Density function of the vote probability distribution for two non-parallel segments.

as $\int_{s\in I_a} \delta(x - g_{ax}(s))\delta(y - g_{ay}(s))\,\mathrm{d}\,s$, and s_b analogously. Then, we can write the Radon transform of s_a with respect to s_b in a following way:

$$\begin{aligned}\mathcal{R}_{s_b}(s_a, t) &= \int_{(x,y)\in s_b+t} \int_{q\in I_a} \delta(x - g_{ax}(q))\delta(y - g_{ay}(q))\,\mathrm{d}\,q\,\mathrm{d}\,x\,\mathrm{d}\,y \\ &= \int_{s\in I_b} \int_{q\in I_a} \delta(g_{bx}(s) + t_x - g_{ax}(q))\delta(g_{by}(s) + t_y - g_{ay}(q))\,\mathrm{d}\,q\,\mathrm{d}\,s\end{aligned}$$

Since the segments are parallel the integral evaluates to zero for all t that do not lie on the segment $T_b = s_a \oplus (-s_b)$. For a translation $t \in T_b$ and a fixed value of s, that is, a fixed point on the translated segment s_b, there exists at most one value of q (representing a point on s_a that coincides with the point $g_b(s) + t$) such that the arguments of the δ-functions evaluate to zero. These s, q-pairs correspond to the points where the segments s_a and $s_b + t$ overlap. Thus, the Radon transform for $t \in T_b$ is

$$\mathcal{R}_{s_b}(s_a, t) = |s_a \cap (s_b + t)|\,\delta((k_a - k_b) - t_x \cos\theta_a - t_y \sin\theta_a)\ ,$$

where $|s_a \cap (s_b + t)|$ denotes the length of the overlapping parts and $\delta((k_a - k_b) - t_x \cos\theta - t_y \sin\theta)$ constrains the vectors t to the line containing T_b. The density function in this case is illustrated in Figure 4.2.

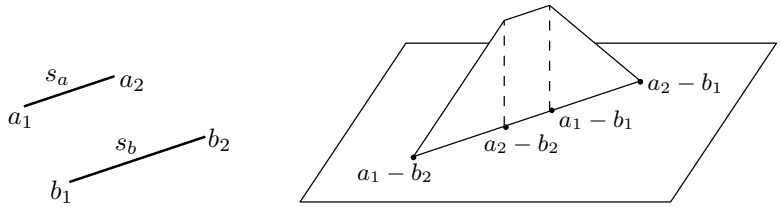


Figure 4.2: Density function of the vote probability distribution for two parallel segments.

Now if our shapes A and B both consist of a finite number of straight line segments, then the density function of the corresponding distribution in the translation space

is the weighted sum over all pairs of segments and can be written as

$$f_T(t) = \sum_{s_a \in A} \sum_{s_b \in B} \frac{|s_a|\,|s_b|}{|A|\,|B|} \frac{1}{|s_a|\,|s_b|} \mathcal{R}_{s_b}(s_a, t)$$
$$= \frac{1}{|A|\,|B|} \sum_{s_a \in A} \sum_{s_b \in B} \mathcal{R}_{s_b}(s_a, t).$$

That is, the support of the density function is composed of all pairwise regions, and if two or more regions overlap, the values of the density function in the intersection region is the sum of the corresponding Radon transform values, see Figure 4.3

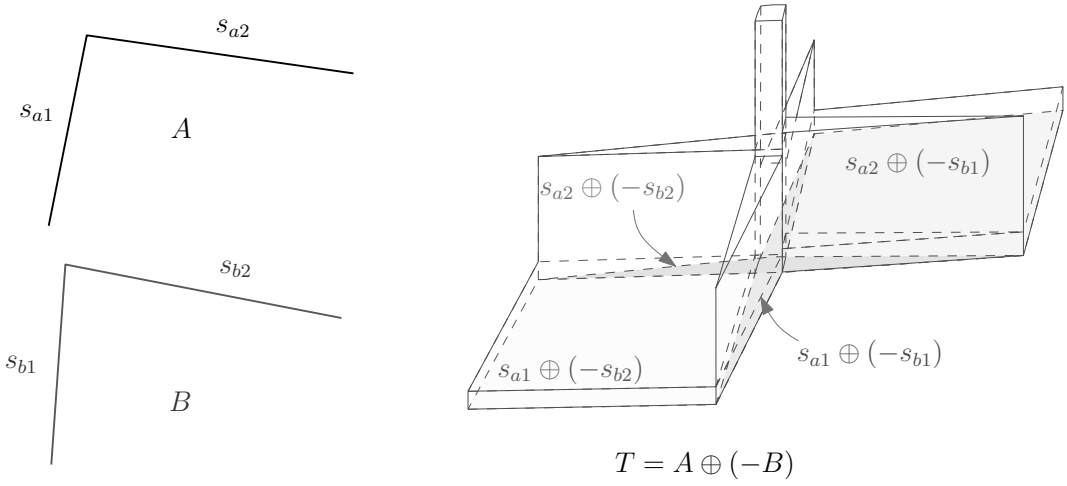


Figure 4.3: Density function in the translation space for two shapes A and B each consisting of two straight line segments.

By considering the arrangement of the δ-regions of votes in the translation space we perform a so-called kernel-based density estimation, see [54]. The kernel-function is in this case a "box"-function which is $1/N$ on a δ-region and zero elsewhere, where N is the number of δ-regions.

Transformation of random variables. In the above paragraph we gave a geometric reasoning for the closed form of the Radon transform for straight line segments. Here we show that the same expression for the density function can be derived from the transformation of random variables. Consider parameterization of segments given by Equation (4.2). If s_a and s_b are not parallel, then the distribution of votes in translation space is non-zero on a parallelogram $T = s_a \oplus (-s_b)$ and the function $\varphi : I_a \times I_b \to T$ with $\varphi(s, r) = g_a(s) - g_b(r)$ is a bijection. The uniform distribution of points on segments s_a and s_b corresponds to the uniform distribution of parameters s and r over the interval I_a and I_b, respectively. The density function on $\mathcal{T}$ under

transformation φ is defined as

$$f_\varphi(v_x, v_y) = f_{s,r}(s,r) \frac{1}{\left| \frac{\partial(v_x, v_y)}{\partial(s,r)} \right|},$$

where $f_{s,r}(s,r)$ denotes the joint density function of the parameters s and r and

$$\begin{aligned}
\frac{\partial(v_x, v_y)}{\partial(s,r)} &= \det \begin{pmatrix} g'_{ax}(s) & -g'_{bx}(r) \\ g'_{ay}(s) & -g'_{by}(r) \end{pmatrix} \\
&= \begin{vmatrix} -\sin\theta_a & \sin\theta_b \\ \cos\theta_a & -\cos\theta_b \end{vmatrix} \\
&= \sin\theta_a \cos\theta_b - \cos\theta_a \sin\theta_b \\
&= \sin(\theta_a - \theta_b) \ .
\end{aligned}$$

Observe that in the parametric representation given by Equation (4.2) the length of a segment corresponds to the length of the parameter interval which defines the segment. Therefore, and since the parameters s and r are independent, the density function on $\mathcal{T}$ is

$$f_\varphi(v_x, v_y) = \frac{1}{|s_a|\,|s_b|} \frac{1}{|\sin(\theta_a - \theta_b)|} \ .$$

If the segments s_a, s_b are parallel, that is, $\theta_a = \theta_b = \theta$, then $T = s_a \oplus (-s_b)$ is a segment in $\mathbb{R}^2$, with parametric representation

$$\begin{aligned}
g_{Tx}(q) &= (k_a - k_b)\cos\theta - q\sin\theta \\
g_{Ty}(q) &= (k_a - k_b)\sin\theta - q\cos\theta \ ,
\end{aligned}$$

where $q = s - r$ and if $I_a = [s_0, s_1], I_b = [r_0, r_1]$ then $q \in [s_0 - r_1, s_1 - r_0] = I_T$. In this case the distribution of votes over $\mathcal{T}$ corresponds to the distribution of parameter values q over I_T. q is the sum of two independent random variables s and $-r$ and has the density function

$$\begin{aligned}
f_q(q) &= \int_{-\infty}^{\infty} f_s(q+r) f_r(r)\,\mathrm{d}\,r \\
&= \frac{1}{|s_b|} \int_{I_b} f_s(q+r)\,\mathrm{d}\,r && \text{since } f_r(r) = \frac{1}{|s_b|} \text{ on } I_b \text{ and 0 else} \\
&= \frac{1}{|s_b|} \int_{I_b \cap I_a - q} \frac{1}{|s_a|}\,\mathrm{d}\,r && \text{since } f_s(s) = \frac{1}{|s_a|} \text{ on } I_a \text{ and 0 else} \\
&= \frac{1}{|s_a|\,|s_b|} \left| I_b \cap (I_a - q) \right| \\
&= \frac{1}{|s_a|\,|s_b|} \left| (I_b + q) \cap I_a \right| \ .
\end{aligned}$$

As we have already mentioned, the length of the parameter interval corresponds to the length of the segment defined by this interval. The value of the density function for the parameter q is, therefore, proportional to the length of the overlapping segment induced by q.

The density function in two-dimensional translation space is then:

$$f_T(t_x, t_y) = \frac{1}{|s_a|\,|s_b|}\,|s_a \cap (s_b + t)|\,\delta((k_a - k_b) - t_x\cos\theta - t_y\sin\theta) \ .$$

Observe that the density function can be integrated over a two-dimensional region for non-parallel as well as for parallel segments.

4.2 Rigid Motions

Random rotation angle

We first consider the variant of taking a random point of each shape and a random rotation angle to determine a rigid motion. We show that in this case the density function of the probability distribution in transformation space is the weighted generalized Radon transform and derive a closed formula to compute it.

We introduce a new random variable X_r corresponding to the rotation angle with density function $f_r(\alpha) = \frac{1}{2\pi}$. The transformation space is now $T = \mathbb{R}^2 \times [0, 2\pi]$. Let $P = V \times I$ denote a region in $\mathcal{T}$ where $V \subset \mathbb{R}^2$ and $I \subset [0, 2\pi]$, the probability distribution on $\mathcal{T}$ is then:

$$\begin{aligned}
F(P) &= \Pr(p \in P) = \Pr(X_r \in I, X_A - X_r(X_B) \in V) \\
&= \int_{\alpha\in I} f_r(\alpha)\Pr(X_A - \alpha(X_B) \in V)\,\mathrm{d}\,\alpha \\
&= \int_{\alpha\in I} \frac{1}{2\pi}\int_{v\in V}\int_{x\in\mathbb{R}^2} f_A(v(\alpha(x)))f_B(x)\,\mathrm{d}\,x\,\mathrm{d}\,v\,\mathrm{d}\,\alpha \\
&= \frac{1}{2\pi\,|A|\,|B|}\int_{\alpha\in I}\int_{v\in V}\int_{x\in\mathbb{R}^2} \delta(C_A(x))\delta(C_B(\alpha(x), v))\,\mathrm{d}\,x\,\mathrm{d}\,v\,\mathrm{d}\,\alpha \\
&= \frac{1}{2\pi\,|A|\,|B|}\int_{p\in P}\int_{x\in\mathbb{R}^2} \delta(C_A(x))\delta(C_B(x, p))\,\mathrm{d}\,x\,\mathrm{d}\,p \\
&= \frac{1}{2\pi\,|A|\,|B|}\int_{p\in P} \mathcal{R}_B(I_A, p)\,\mathrm{d}\,p \qquad (4.3)
\end{aligned}$$

Closed formula. In order to find a closed form of this density function we again first consider A and B consisting of a single segment s_a and s_b respectively, in parameter representation as given by (4.2). Note that the parameter form of the line supporting segment s_b under rotation α has following form:

$$\begin{aligned}
g_{bx}(r) &= k_a\cos(\theta_b + \alpha) - r\sin(\theta_b + \alpha) \\
g_{by}(r) &= k_a\sin(\theta_b + \alpha) + r\cos(\theta_b + \alpha) \ .
\end{aligned}$$

The function $\varphi : [0, 2\pi) \times \mathbb{R}^2 \to [0, 2\pi) \times \mathbb{R}^2$ with $\varphi(\alpha, s, r) = (\alpha, g_a(s) - M_\alpha g_b(r))$ is a bijection except for two values of α ($\theta_a - \theta_b$ and $\theta_a - \theta_b + \pi$) which makes the two segments be parallel. Now with these two values excluded and with similar computations as for the case of translations we find that the density function in the space of rigid motions is

$$f_T(\alpha, v) = \frac{1}{2\pi\,|s_a|\,|s_b|} \cdot \frac{1}{|\sin(\theta_a - \theta_b - \alpha)|} \ .$$

For the two values of α, for which the rotated segment s_b is parallel to the segment s_a the computations are again similar to those in case of translations, which gives us the following density function:

$$f_T(\alpha, v_x, v_y) = \frac{1}{2\pi\,|s_a|\,|s_b|}\,|s_a \cap (M_\alpha s_b + v)|\,\delta((k_a - k_b) - v_x\cos\theta_a - v_y\sin\theta_a) \ .$$

Note that in this case we have a delta-function defined only in the translation planes corresponding to some special values of α. Putting it together we get:

$$f_T(\alpha, v) = \begin{cases} \frac{1}{2\pi|s_a||s_b|} \cdot \frac{1}{|\sin(\theta_a-\theta_b-\alpha)|} \ , & \text{if } \alpha \notin \{\theta_a - \theta_b, \theta_a - \theta_b + \pi\} \\ & \text{and } v \in s_a \oplus (-M_\alpha s_b) \\ \frac{1}{2\pi|s_a||s_b|}\,|s_a \cap (M_\alpha s_b + v)|\,\delta((k_a - k_b) - v_x\cos\theta_a - v_y\sin\theta_a) \ , & \\ & \text{if } \alpha \in \{\theta_a - \theta_b, \theta_a - \theta_b + \pi\} \\ 0 \ , & \text{else.} \end{cases}$$

Rotation angle determined by direction of tangent lines

In this case a random experiment consists of taking a random point from each shape $a \in A, b \in B$ and determining the angle between the tangent line to the curve at the selected point and the x-axis. Since we consider the curves to be undirected, we also consider the tangent lines to be undirected. Then there are exactly two rotation angles that transform the tangent at the point b to a line parallel to the tangent at the point a. So there are exactly two rigid motions that map the point b to the point a so that their tangents are parallel. The rotation angles α_1, α_2 are determined from the difference of slopes of the tangent lines and the translation vectors v_1, v_2 are computed as $v_i = a - M_{\alpha_i} b$, where $i \in \{1, 2\}$ and M_{α_i} is the rotation matrix corresponding to the angle α_i.

If A and B both consist of polygonal curves and the tangent directions are not interpolated, then, obviously, there are finitely many directions of tangents and, thus, rotation angle α can have finitely many values. The probability for a certain value of α to occur equals the probability of selecting points on the "right" segments, that is, the segments that have difference in tangent angles exactly α.

For the formal definition we introduce some additional notation: Let $s_{a1}, \ldots, s_{an}$ be the segments contained in shape A and $s_{b1}, \ldots, s_{bm}$ the segments of shape B. Further, let L_A, L_B denote the total length of curves in A and B respectively, L_{ai}

denotes the length of the segment s_{ai} and L_{bj} the length of s_{bj}. Further, let θ_{ai}, k_{ai} and θ_{bj}, k_{bj} denote the parameters of the normal form of the lines containing the segments s_{ai} and s_{bj}, respectively. The density of the distribution of rotation angles can be expressed as

$$f_\alpha(\alpha) = \sum_{\substack{1\le i\le n\\ 1\le j\le m}} \frac{L_{ai}L_{bj}}{L_A L_B}\delta(\theta_{ai}-\theta_{bj}-\alpha) + \sum_{\substack{1\le i\le n\\ 1\le j\le m}} \frac{L_{ai}L_{bj}}{L_A L_B}\delta(\theta_{ai}+\pi-\theta_{bj}-\alpha) \ ,$$

where δ is Dirac's delta-function.

For each fixed value of α that has non-zero probability we get only the votes from the segments that are parallel if B is rotated by α. As we have seen above, these votes are distributed along a line segment in the corresponding translation plane. Combining these two observations we get that the votes in three dimensional space of rigid motions are distributed along at most $2 \cdot n \cdot m$ straight line segments and have the following density function:

$$f_T(\alpha, v) = \frac{1}{L_A L_B} \sum_{\substack{1\le i\le n\\ 1\le j\le m}} |s_{ai} \cap (M_\alpha s_{bj} + v)| \cdot \delta(k_{ai} - k_{bj} - v_x \cos\theta_{ai} - v_y \sin\theta_{ai}) \cdot (\delta(\theta_{ai} - \theta_{bj} - \alpha) + \delta(\theta_{ai} + \pi - \theta_{bj} - \alpha)) \ .$$

4.3 Distribution of the Scaling Factor

It is an intuitive technique to reduce the dimensionality of the space in consideration, e.g., space of the similarity transformations, by considering the dimensions separately. So one could first determine the "best" scaling factor k for two given shapes A and B. Then the shape B is scaled by the factor k and the shapes A and $k \cdot B$ are matched under rigid motions. The matching under rigid motions might again be split into determining the rotation angle separately and then matching under translations.

In this section we consider the distribution of the scaling factor in the following random experiment: Given two shapes A and B, two points from each shape $a_1, a_2 \in A$, $b_1, b_2 \in B$ are chosen randomly under uniform distribution with respect to length. With every four such points we associate a random variable $X_s = \frac{\mathrm{dist}(a_1,a_2)}{\mathrm{dist}(b_1,b_2)}$, where $\mathrm{dist}(\cdot,\cdot)$ denotes the Euclidean distance between two points. That is, X_s is the scaling factor that would map the segment b_1b_2 to a segment of length equal to the length of a_1a_2.

Let $X_{da} = \mathrm{dist}(a_1, a_2)$ and $X_{db} = \mathrm{dist}(b_1, b_2)$ denote the random variables associated with the distance between two randomly generated points on each of the segments and f_{da}, f_{db} the corresponding density functions. Note that X_{da} and X_{db} are independent and their joint density function is, therefore, a product of the individual density functions.

Then $X_s = \frac{X_{da}}{X_{db}}$ and the density function f_s of X_s is defined by

$$f_s(z) = \int_{-\infty}^{\infty} |x|\, f_{da}(zx) f_{db}(x)\, \mathrm{d}\, x \ . \tag{4.4}$$

This follows from a more general statement in the following Lemma:

Lemma 4.1. *Let X, Y be continuous random variables with joint density function f_{XY}, then the density function of $Z = X/Y$ is*

$$f_Z(z) = \int_{-\infty}^{\infty} |y|\, f_{XY}(yz, y)\, \mathrm{d}\, y \quad \text{for } z \in \mathbb{R} \ .$$

Proof. For jointly continuous variables X, Y and any regular subset A of $\mathbb{R}^2$ it is known that $P((X,Y) \in A) = \iint_A f_{XY}(x,y)\, \mathrm{d}\, x\, \mathrm{d}\, y$ (see e.g. [32, Theorem 6A]). Consider the region A_z defined as the set of (x, y) such that $x/y \leq z$, that is, $A_z = \{(x,y) \in \mathbb{R}^2 : x/y \leq z\}$. The condition $x/y \leq z$ is equivalent to $x \leq yz$ for the positive values of y and to $x \geq yz$ for the negative y as depicted in Figure 4.4.

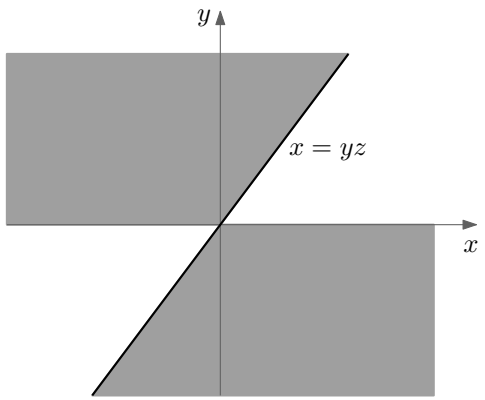


Figure 4.4: Shaded region denotes A_z, drawn for the case $z > 0$.

Then the distribution function of Z is defined by

$$F_Z(z) = \int_0^{\infty} \int_{-\infty}^{yz} f_{XY}(x,y)\, \mathrm{d}\, x\, \mathrm{d}\, y + \int_{-\infty}^{0} \int_{yz}^{\infty} f_{XY}(x,y)\, \mathrm{d}\, x\, \mathrm{d}\, y \ .$$

The density f_Z is found by differentiating F_Z to obtain

$$\begin{aligned} f_Z(z) = \frac{\mathrm{d}}{\mathrm{d}\, z} F_Z(z) &= \int_0^{\infty} y f_{XY}(yz, y)\, \mathrm{d}\, y + \int_{-\infty}^{0} -y f_{XY}(yz, y)\, \mathrm{d}\, y \\ &= \int_{-\infty}^{\infty} |y|\, f_{XY}(yz, y)\, \mathrm{d}\, y \ . \end{aligned}$$

□

The distribution of the scaling factor obviously depends on the distribution of point distances on the shapes. In the following we examine this distribution for some basic cases.

4.3.1 Distance Distribution on a Line Segment

Here we examine the distribution of the point distances on a line segment. The corresponding random experiment is the following: Given a segment s of length L, two points are selected randomly on s under uniform distribution with respect to length, and the distance between these points is assigned to the random variable X_d.

Observe that this random experiment is equivalent to selecting two real numbers x_1, x_2 in the interval $[0, L]$ and computing the absolute difference between them $X_d = |x_1 - x_2|$. The numbers x_1, x_2 correspond to the parameter values in the natural parameterization of a segment, and the absolute difference of the parameter values is exactly the Euclidean distance between the corresponding points. The density function $f(x)$ of the numbers x_1, x_2 equals to $\frac{1}{L}$ for $x \in [0, L]$ and zero elsewhere.

Consider first the distribution of the difference $X_{d'} = x_1 - x_2$. By the formula for the sum of random variables we get:

$$f_{d'}(z) = -\int_{-\infty}^{\infty} f(x) f(x - z)\,\mathrm{d}\,x = \int_{\max(0,z)}^{\min(L,L+z)} \frac{1}{L^2}\,\mathrm{d}\,x \ ,$$

which resolves into a symmetric function

$$f_{d'}(z) = \begin{cases} \frac{L-z}{L^2} & \text{for } 0 \le z \le L \\ \frac{L+z}{L^2} & \text{for } -L \le z \le 0 \\ 0 & \text{else} \end{cases} .$$

Since $X_d = |X_{d'}|$, the density of X_d is $f_d(z) = f_{d'}(z) + f_{d'}(-z) = 2f_{d'}(z) = 2\frac{L-z}{L^2}$ if $0 \le z \le L$ and zero otherwise. Thus, the density of the distance variable X_d is

$$f_d(x) = \begin{cases} 2\frac{L-x}{L^2} & \text{for } 0 \le x \le L \\ 0 & \text{else} \end{cases} . \tag{4.5}$$

The expected value of the distance variable X_d is $\mathrm{E}(X_d) = \frac{1}{3}L$.

4.3.2 Distribution of the Scaling Factor for two Line Segments.

Let the shapes A and B consist of a single straight line segment of length L_a and L_b respectively. Then the density functions of the random variables denoting the distance between two random points of each shape are according to Equation (4.5):

$$f_{da}(x) = \begin{cases} 2\frac{L_a-x}{L_a^2} & \text{for } 0 \le x \le L_a \\ 0 & \text{else} \end{cases} \qquad f_{db}(x) = \begin{cases} 2\frac{L_b-x}{L_b^2} & \text{for } 0 \le x \le L_b \\ 0 & \text{else} \end{cases} .$$

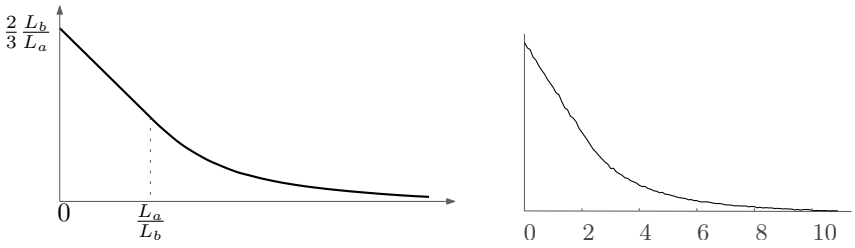
To compute the distribution of the scaling factor variable X_s we can plug these function into the equation (4.4):

$$\begin{aligned} f_s(z) &= \int_{-\infty}^{\infty} |x|\, f_{da}(zx) f_{db}(x)\,\mathrm{d}\,x \\ &= \frac{4}{L_a^2 L_b^2} \int_0^{\min(L_b, L_a/z)} x(L_a - zx)(L_b - x)\,\mathrm{d}\,x \ . \end{aligned}$$

The expression has to be integrated separately for $0 \leq z \leq \frac{L_a}{L_b}$ and for $z > \frac{L_a}{L_b}$:

$$\begin{aligned} \text{If } 0 \leq z \leq \frac{L_a}{L_b}: \qquad f_s(z) &= \frac{4}{L_a^2 L_b^2} \int_0^{L_b} x(L_a - zx)(L_b - x)\,\mathrm{d}\,x \\ &= -\frac{z L_b^2}{3 L_a^2} + \frac{2 L_b}{3 L_a} \ . \\ \text{If } z \geq \frac{L_a}{L_b}: \qquad f_s(z) &= \frac{4}{L_a^2 L_b^2} \int_0^{L_a/z} x(L_a - zx)(L_b - x)\,\mathrm{d}\,x \\ &= -\frac{L_a^2}{3 z^3 L_b^2} + \frac{2 L_a}{3 z^2 L_b} \ . \end{aligned}$$

The plot of the function f_s is depicted in Figure 4.5(a).



(a) Plot of the density function.

(b) Experimentally constructed density function.

Figure 4.5: Density function of the random variable corresponding to the scaling factor.

The expectation value of X_s does not exist since the integral $\int_0^{\infty} z \cdot f_s(z)\,\mathrm{d}\,z$ does not evaluate to a real value.

Since the density function $f_s(z)$ of the scaling factor distribution attains its maximum at $z = 0$ the described experiment is not suitable for the computation of the "best" scaling factor for two line segments separately from rotation and translation. By the "best" scaling factor for two line segments we mean the factor of the scaling transformation that transforms the segment B to a segment of the length L_a, that is $z = \frac{L_a}{L_b}$. Thus, for the purpose of being able to determine the appropriate scaling factor for two figures from the described experiment, we need to modify this function in a way that penalizes small scaling factors. One function with this property is

$f(z) = z \cdot f_s(z)$, see Figure 4.6. The intuition behind the choice of the function $f(z)$ is that the multiplicative factor z penalizes small scaling factors directly, proportionally to the value of the scaling factor. Additionally, since the function $f_s(z)$ decreases rapidly for large values of z, multiplication with z does not overweight large scaling factors. Namely, for large values of z the function remains monotone decreasing. In fact, the function $f(z)$ has exactly one maximum at $z = \frac{L_a}{L_b}$: The derivative

$$f'(z) = \begin{cases} -\frac{2L_b}{3L_a^2}\left(zL_b - L_a\right), & 0 \le z \le \frac{L_a}{L_b} \\ -\frac{2L_a}{3L_b^2}\frac{(zL_b - L_a)}{z^3}, & z > \frac{L_a}{L_b} \end{cases}$$

is zero for $z = \frac{L_a}{L_b}$, and the second derivative at that point is negative.

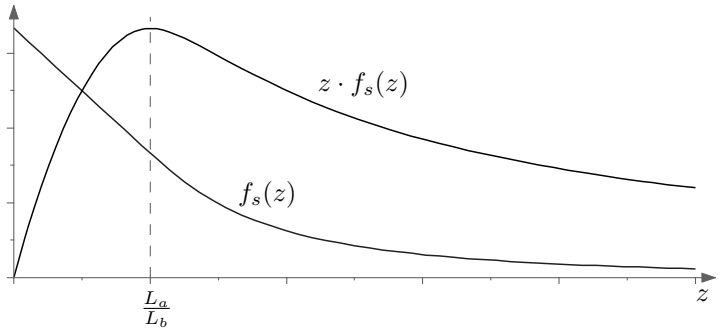


Figure 4.6: Graphs of the density function $f_s(z)$ of the scaling distribution for two line segments and of the function $z \cdot f_s(z)$. The function $z \cdot f_s(z)$ attains its maximum at $z = \frac{L_a}{L_b}$.

The modified scale distribution function is easy to approximate experimentally with standard binning techniques and it performs well on the MPEG-7 Shape B dataset (see Section A.2) in sense that the scaling factor where the modified distribution function attains its maximum is the factor that makes the shapes approximately equally large.

4.3.3 Distance Distribution on a Circle

Selecting a point randomly uniformly distributed on a circle of radius r corresponds to selecting an angle φ uniformly randomly from the interval $[0, 2\pi)$. The corresponding point is then $(r\cos\varphi, r\sin\varphi)$. So if we want to select two points randomly on a circle, we need to select two angles φ_1, φ_2 independently randomly uniformly distributed from the interval $[0, 2\pi)$. The density function of the angle variables φ_1, φ_2 is $f(x) = \frac{1}{2\pi}$ for $x \in [0, 2\pi)$ end 0 elsewhere. The distance between the corresponding points

$p = (r\cos\varphi_1, r\sin\varphi_1)$ and $q = (r\cos\varphi_2, r\sin\varphi_2)$ is

$$\begin{aligned}\operatorname{dist}(\varphi_1,\varphi_2) &= r\sqrt{(cos\varphi_1 - cos\varphi_2)^2 + (\sin\varphi_1 - \sin\varphi_2)^2} \\ &= r\sqrt{2 - 2(\cos\varphi_1\cos\varphi_2 + \sin\varphi_1\sin\varphi_2)} \\ &= r\sqrt{2 - 2\cos(\varphi_1 - \varphi_2)} \ .\end{aligned}$$

In the following we examine the distribution of the distance random variable $X_d = \operatorname{dist}(\varphi_1, \varphi_2)$.

Let $g(z)$ denote the maximal absolute difference that the angles φ_1, φ_2 can have so that the distance between points p and q is at most z. Obviously we are only interested in $0 \le z \le 2r$. Since $\operatorname{dist}(\varphi_1,\varphi_2) \le z$ iff $\operatorname{dist}^2(\varphi_1,\varphi_2) \le z^2$, we get that

$$g(z) = \arccos\left(1 - \frac{z^2}{2r^2}\right) \ .$$

The probability that the distance between two randomly drawn points is at most z, that is, the distribution function of the random variable X_d, can be expressed as

$$F_d(z) = \Pr(X_d \le z) = \int_{\varphi_1=0}^{2\pi} \int_{\varphi_2=\varphi_1-g(z)}^{\varphi_1+g(z)} f(\varphi_1) f(\varphi_2)\,\mathrm{d}\,\varphi_2\,\mathrm{d}\,\varphi_1 \ ,$$

where the addition and subtraction operation in the integral boundaries are meant to be performed modulo interval boundaries of the interval $[0, 2\pi]$. The density function can be found as

$$\begin{aligned} f_d(z) &= \frac{\mathrm{d}}{\mathrm{d}\,z} F_d(z) \\ &= \int_{\varphi_1=0}^{2\pi} \frac{\partial}{\partial z} \left(\int_{\varphi_2=\varphi_1-g(z)}^{\varphi_1+g(z)} f(\varphi_1) f(\varphi_2)\,\mathrm{d}\,\varphi_2 \right) \mathrm{d}\,\varphi_1 \quad \text{(Leibniz integral rule)} \\ &= \int_{\varphi_1=0}^{2\pi} \Bigg(g'(z)\cdot f(\varphi_1) f(\varphi_1 + g(z)) - (-g'(z))\cdot f(\varphi_1) f(\varphi_1 - g(z)) \\ &\qquad + \int_{\varphi_2=\varphi_1-g(z)}^{\varphi_1+g(z)} \frac{\partial}{\partial z}\left[f(\varphi_1) f(\varphi_2)\right] \mathrm{d}\,\varphi_2 \Bigg) \mathrm{d}\,\varphi_1 \\ &\qquad\qquad \text{(generalization of Leibniz integral rule, see [38, p. 275])} \\ &= \int_{\varphi_1=0}^{2\pi} g'(z)\cdot f(\varphi_1)\left(f(\varphi_1 + g(z)) + f(\varphi_1 - g(z))\right) \mathrm{d}\,\varphi_1 \\ &= \frac{1}{2\pi^2} \int_{\varphi_1=0}^{2\pi} g'(z)\,\mathrm{d}\,\varphi_1 = \frac{1}{2\pi^2}\ g'(z)\cdot\varphi_1\Big|_{\varphi_1=0}^{2\pi} \\ &= \frac{2}{\pi\sqrt{4r^2 - z^2}} \ . \end{aligned} \tag{4.6}$$

The function has a pole at $z = 2r$. The graph of $f_d(z)$ is depicted in Figure 4.7. The expectation value of the distance variable X_d for two points on a circle of radius r is $\mathrm{E}(X_d) = \frac{4}{\pi} r$.

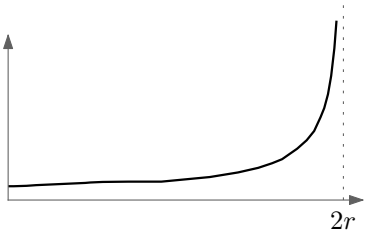


Figure 4.7: Graph of the distance distribution density function for two points on a circle.

4.3.4 Distribution of the Scaling Factor for two Circles.

Let the shapes A and B be two circles of radii r_a and r_b respectively. Then the density functions of the distance between two randomly chosen points on each shape are given by Formula (4.6):

$$f_{da}(x) = \begin{cases} \frac{2}{\pi\sqrt{4r_a^2 - x^2}} & \text{for } 0 \le x \le 2r_a, \\ 0 & \text{else,} \end{cases} \qquad f_{db}(x) = \begin{cases} \frac{2}{\pi\sqrt{4r_b^2 - x^2}} & \text{for } 0 \le x \le 2r_b, \\ 0 & \text{else.} \end{cases}$$

The density function of the scale variable X_s is then according to equation (4.4):

$$\begin{aligned} f_s(z) &= \int_{-\infty}^{\infty} |x|\, f_{da}(zx) f_{db}(x)\, \mathrm{d}\, x \\ &= \frac{4}{\pi^2} \int_0^{\min\{2r_b, 2r_a/z\}} \frac{x}{\sqrt{(4r_a^2 - z^2x^2)(4r_b^2 - x^2)}}\, \mathrm{d}\, x \end{aligned}$$

with variable substitution $y = x^2$ and $\mathrm{d}\, y = 2x\, \mathrm{d}\, x$ we get

$$\begin{aligned} &= \frac{2}{\pi^2} \int_0^{(\min\{2r_b, 2r_a/z\})^2} \frac{1}{\sqrt{(4r_a^2 - z^2y)(4r_b^2 - y)}}\, \mathrm{d}\, y \\ &= \frac{2}{\pi^2} \int_0^{(\min\{2r_b, 2r_a/z\})^2} \frac{1}{\sqrt{z^2y^2 - y(4r_b^2z^2 + 4r_a^2) + 16r_a^2r_b^2}}\, \mathrm{d}\, y \end{aligned}$$

by completing the square we can further simplify the expression:

$$= \frac{2}{\pi^2 z} \int_0^{(\min\{2r_b, 2r_a/z\})^2} \frac{1}{\sqrt{\left(y - \frac{2(r_b^2z^2 + r_a^2)}{z^2}\right)^2 - \left(\frac{2(r_b^2z^2 - r_a^2)}{z^2}\right)^2}}\, \mathrm{d}\, y$$

now we perform another variable substitution $u = y - \frac{2(r_b^2z^2 + r_a^2)}{z^2}$:

$$= \frac{2}{\pi^2 z} \int_{-\frac{2(r_b^2z^2 + r_a^2)}{z^2}}^{(\min\{2r_b, 2r_a/z\})^2 - \frac{2(r_b^2z^2 + r_a^2)}{z^2}} \frac{1}{\sqrt{u^2 - \left(\frac{2(r_b^2z^2 - r_a^2)}{z^2}\right)^2}}\, \mathrm{d}\, y$$

$$= \frac{2}{\pi^2 z}\left(\ln\left|u+\sqrt{u^2-\left(\frac{2(r_b^2z^2-r_a^2)}{z^2}\right)^2}\right|\Bigg|_{-\frac{2(r_b^2z^2+r_a^2)}{z^2}}^{(\min\{2r_b,2r_a/z\})^2-\frac{2(r_b^2z^2+r_a^2)}{z^2}}\right) .$$

For $z \in [0, r_a/r_b]$ it holds that $r_b \leq r_a/z$ and the function $f_s(z)$ evaluates to

$$\begin{aligned} f_s(z) &= \frac{2}{\pi^2 z}\left(\ln\left(-\frac{2(r_b^2z^2-r_a^2)}{z^2}\right)-\ln\left(\frac{2(r_bz-r_a)^2}{z^2}\right)\right) \\ &= \frac{2}{\pi^2 z}\ln\left(\frac{r_a+r_bz}{r_a-r_bz}\right) . \end{aligned}$$

For $z > r_a/r_b$ we have that $r_a/z < r_b$ and

$$\begin{aligned} f_s(z) &= \frac{2}{\pi^2 z}\left(\ln\left(\frac{2(r_b^2z^2-r_a^2)}{z^2}\right)-\ln\left(\frac{2(r_bz-r_a)^2}{z^2}\right)\right) \\ &= \frac{2}{\pi^2 z}\ln\left(\frac{r_bz+r_a}{r_bz-r_a}\right) . \end{aligned}$$

Summarising the above computations we get: Given two circles A of radius r_a and B of radius r_b, we consider the random experiment where two points on each shape are chosen randomly uniformly distributed with respect to length. The value of the scaling factor that makes the distance between the points of the shape B equal to the distance between the points of the shape A is recorded in the random variable X_s. The density function of the variable X_s is (see Figure 4.8 for an illustration)

$$f_s(z) = \begin{cases} 0 & \text{for } z < 0, \\ \frac{2}{\pi^2 z}\ln\left(\frac{r_a+r_bz}{r_a-r_bz}\right) & 0 \leq z \leq \frac{r_a}{r_b}, \\ \frac{2}{\pi^2 z}\ln\left(\frac{r_bz+r_a}{r_bz-r_a}\right) & z > \frac{r_a}{r_b} . \end{cases}$$

For z approaching zero from the right the limit of the function f_s is according to de l'Hospital's rule

$$\lim_{z\to 0^+} f_s(z) = \lim_{z\to 0^+} \frac{2}{\pi^2}\cdot\frac{2r_ar_b}{r_a^2-r_b^2z^2} = \frac{4r_b}{\pi^2 r_a} .$$

The function has a pole at $z = r_a/r_b$, which is exactly the scaling factor that maps the radius of the circle B to the radius of the circle A.

Note that the modification to the distribution function $f(z) = z \cdot f_s(z)$ introduced in the Section 4.3.2, when applied to the scaling distribution function for two circles, also has a pole at $z = \frac{r_a}{r_b}$ and does not introduce any new local maxima.

4.3.5 Distance Distribution on two Line Segments

Given a shape A consisting of two segments s_1 and s_2 with lengths l_1, l_2 and total length $L = l_1 + l_2$, consider the following random experiment: Select randomly,

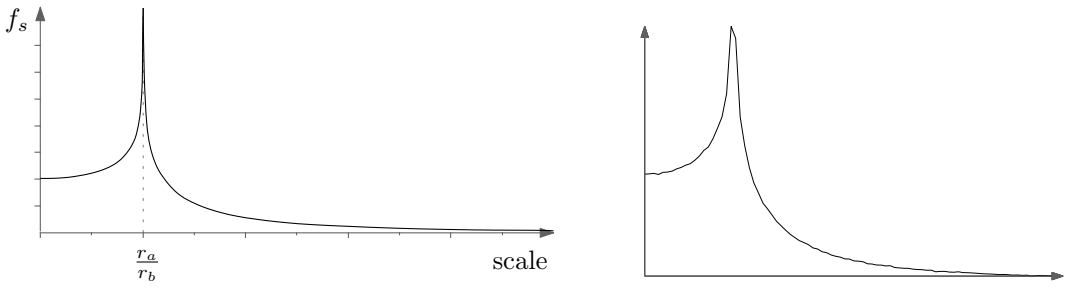


Figure 4.8: Graph of the density function of the scale distribution for two circles plotted by MAPLE in the fist figure and computed experimentally in the second.

independently, uniformly distributed with respect to length two points on A and assign the distance between these points to the random variable X_d.

The randomly selected points can either be both on the same segment, or on two different segments. Thus, the density function $f_d(x)$ of the random variable X_d is a weighted sum of the density functions for each of the two cases:

$$f_d(x) = \sum_{i=1,2} \frac{l_i^2}{L^2} f_{d1}(x, s_i) + 2\frac{l_1 \cdot l_2}{L^2} f_{d2}(x, s_1, s_2) \ , \tag{4.7}$$

where $f_{d1}(x, s_i)$ denotes the density function of the distance distribution on the segment s_i given by Equation (4.5), and $f_{d2}(x, s_1, s_2)$ denotes the density function of the distance distribution where one random point is located on the segment s_1 and the other point on the segment s_2. The weights $\frac{l_i^2}{L^2}$ and $2\frac{l_1 \cdot l_2}{L^2}$ are the probabilities of the corresponding events. In the following we examine the distribution of the point distances between two segments, that is, the function f_{d2}.

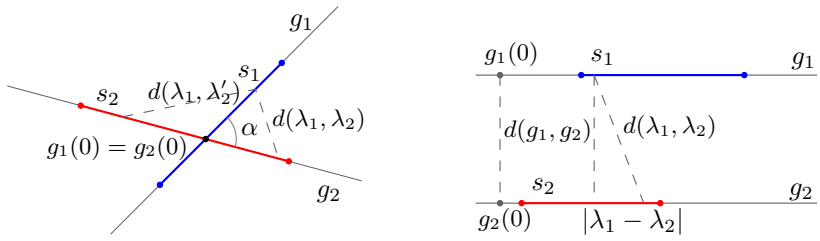
Let g_1, g_2 denote the lines containing segments s_1 and s_2 respectively, and α denote the sharp angle between g_1 and g_2. For simplicity we assume that the lines g_1, g_2 are parameterized by length with the property $g_1(0) = g_2(0)$, that is, their intersection point corresponds to the parameter value zero on both lines, if g_1, g_2 are not parallel, see Figure 4.9(a). If the lines are parallel, choose the position of $g_1(0)$ arbitrarily and the parameterization of g_2 such that $g_2(0)$ is the closest point of g_2 to $g_1(0)$, see Figure 4.9(b). Then the segments s_1 and s_2 correspond to some parameter value intervals $I_1 = [\iota_1^l, \iota_1^u], I_2 = [\iota_2^l, \iota_2^u] \subset \mathbb{R}$, and the squared Euclidean distance between two points $g_1(\lambda_1)$ and $g_2(\lambda_2)$ for some parameter values λ_1, λ_2 is given by

$$d^2(\lambda_1, \lambda_2) = \lambda_1^2 + \lambda_2^2 - 2\lambda_1\lambda_2 \cos\alpha \tag{4.8}$$

for non-parallel lines, and by

$$d^2(\lambda_1, \lambda_2) = (\lambda_1 - \lambda_2)^2 + d^2(g_1, g_2) \tag{4.9}$$

for the parallel lines, where $d(g_1, g_2)$ denotes the distance between the lines.



(a) non-parallel segments (b) parallel segments

Figure 4.9: Two segments s_1 and s_2 and their supporting lines.

For every point $g_1(\lambda_1)$ and the distance value $z \geq 0$ ($z \geq d(g_1, g_2)$ for parallel lines), there exists an interval $I(\lambda_1, z) \subset \mathbb{R}$ such that $\forall \lambda_2 \in I(\lambda_1, z)$ the distance $d(\lambda_1, \lambda_2) \leq z$. The interval $I(\lambda_1, z)$ is

$$I(\lambda_1, z) = [\lambda_1 \cos\alpha - \sqrt{z^2 - \lambda_1^2 \sin^2\alpha},\ \lambda_1 \cos\alpha + \sqrt{z^2 - \lambda_1^2 \sin^2\alpha}] \ ,$$

for non-parallel lines, and

$$I(\lambda_1, z) = [\lambda_1 - \sqrt{z^2 - d^2(g_1, g_2)},\ \lambda_1 + \sqrt{z^2 - d^2(g_1, g_2)}] \ ,$$

for parallel lines. Then the probability that in one random experiment the distance between two points is at most z, that is, the probability distribution function can be written as

$$F_{d2}(z) = \int_{\lambda \in I_1} \int_{\lambda_2 \in I_2'} f_1(\lambda_1) f_2(\lambda_2) \, \mathrm{d}\, \lambda_2 \, \mathrm{d}\, \lambda_1 \ ,$$

where $I_2' = I_2 \cap I(\lambda_1, z)$ and $f_i(\lambda_i)$, $i = 1, 2$, denote the density functions on the segments, i.e., $f_i(\lambda_i) = \frac{1}{l_i}$ for $\lambda_i \in I_i$ and 0 otherwise.

Let $\iota_u''(\lambda_1, z)$ and $\iota_l''(\lambda_1, z)$ denote the upper and lower bounds of the interval I_2'. Since the functions f_1, f_2 are constant over the integration interval we can simplify F_{d2}:

$$F_{d2}(z) = \frac{1}{l_1 l_2} \int_{\lambda \in I_1} \iota_u''(\lambda_1, z) - \iota_l''(\lambda_1, z) \, \mathrm{d}\, \lambda_1 \ .$$

In order to be able to further specify the distribution and the density function, we need to take a closer look at the intervals $I(\lambda_1, z), I_2$ and I_1. Each of the upper and lower bounds of the interval I_2' can be defined by the corresponding bound either of the interval I_2 or the interval $I(\lambda_1, z)$, giving thus four possible cases for the interval bounds plus the possibility of the interval $I(\lambda_1, z)$ being empty:

①. Both upper and lower bounds are defined by $I(\lambda_1, z)$.

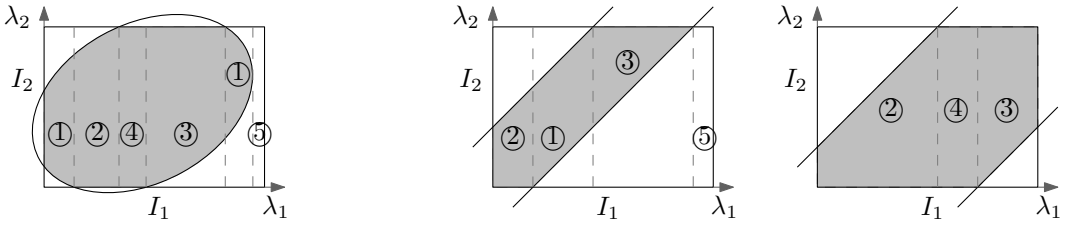
②. The upper bound is defined by $I(\lambda_1, z)$ and the lower bound by I_2.

③. The upper bound is defined by I_2 and the lower bound by $I(\lambda_1, z)$.

④. Both upper and lower bounds are defined by I_2.

⑤. The interval $I(\lambda_1, z)$ is empty.

Note that depending on the distance z and the relative position of the segments, multiple or even all these cases might occur for the values λ_1 in the interval I_1. Examples of the occurrence of each of the described cases are shown in the Figure 4.10. The



(a) non-parallel segments (b) parallel segments

Figure 4.10: An example of the combination of $I_1, I_2, I(\lambda_1, z)$ for non-parallel and parallel segments where each of the described cases occurs over some subinterval of I_1. Shaded regions denote the parameter values (λ_1, λ_2) such that $d(g_1(\lambda_1), g_2(\lambda_2)) \leq z$ and $\lambda_1 \in I_1, \lambda_2 \in I_2$.

function F_{d2} is then a sum of the integrals over the intervals $I_1'(z) = [\iota_l'(z), \iota_u'(z)] \subset I_1$ such that the bounds of I_2' belong to the same case for every $\lambda_1 \in I_1'(z)$. Let $F_{d2}'(z)$ denote the summand of $F_{d2}(z)$ over an interval $I'(z)$. The function F_{d2}' has the form

$$F_{d2}'(z) = \frac{1}{l_1 l_2} \int_{\lambda_1 = \iota_l'(z)}^{\iota_u'(z)} \iota_u''(\lambda_1, z) - \iota_l''(\lambda_1, z) \,\mathrm{d}\,\lambda_1 \ .$$

The density function f_{d2} of the distance distribution is the derivative of the distribution function and is therefore a sum of the derivatives f_{d2}' of the F_{d2}' functions:

$$\begin{aligned} f_{d2}'(z) &= \frac{\mathrm{d}}{\mathrm{d}\,z} F_{d2}'(z) \\ &= \frac{1}{l_1 l_2} \Bigg[\left(\iota_u''(\iota_u'(z), z) - \iota_l''(\iota_u'(z), z) \right) \frac{\mathrm{d}}{\mathrm{d}\,z} \iota_u'(z) \\ &\qquad - \left(\iota_u''(\iota_l'(z), z) - \iota_l''(\iota_l'(z), z) \right) \frac{\mathrm{d}}{\mathrm{d}\,z} \iota_l'(z) \\ &\qquad + \int_{\lambda_1 = \iota_l'(z)}^{\iota_u'(z)} \frac{\partial}{\partial z} \left[\iota_u''(\lambda_1, z) - \iota_l''(\lambda_1, z) \right] \mathrm{d}\,\lambda_1 \Bigg] \ . \end{aligned}$$

In the following we show how the function f_{d2} is determined on the basis of an example of two non-parallel intersecting segments depicted in Figure 4.9(a). As

described above, for every distance value z there is a certain combination of subintervals of I_1 that determine a decomposition of the function f_{d2} into a sum. Furthermore, the order and type of the intervals does not change over a certain interval of the distance values z. We call the values of z, where the combinatorial configuration of the subintervals changes, the *event points*.

It is known and is easy to see from the expression (4.8), that the parameter values (λ_1, λ_2) such that $d(g_1(\lambda_1), g_2(\lambda_2)) \leq z$ form an ellipse $\mathcal{E}(z)$ in the parameter space. The intervals I_1, I_2 define an axis-parallel rectangle. Thus, the points on the segments s_1, s_2 with distance at most z between them correspond to the intersection of the ellipse $\mathcal{E}(z)$ with the rectangle $I_1 \times I_2$. The function $F_{d2}(z)$ is exactly the area of that intersection over the area of the rectangle $I_1 \times I_2$.

The values of z where the left- or rightmost points of $\mathcal{E}(z)$, i.e., points with the lowest or highest λ_1 coordinate, coincide with the lower or upper boundary of the interval I_1 are candidates for the event points, since at that positions the lower or upper boundary of a subinterval changes from the λ_1-coordinate of the corresponding extreme point of $\mathcal{E}(z)$ to the corresponding bound of the interval I_1.

The values of z where the bottom or top-most points of $\mathcal{E}(z)$, i.e., points with the lowest or highest λ_2-coordinate, reach the bounds of the interval I_2, are also candidates for the event points, since at these positions an interval of one type is split into three intervals. The middle one of the three new intervals is of a new type, and is bound by the intersection points of the boundary of the ellipse $\mathcal{E}(z)$ with the lines $\lambda_2 = \iota_2^l$ or $\lambda_2 = \iota_2^u$.

The lowest λ_i-coordinate, $i = 1, 2$, of a point in $\mathcal{E}(z)$ is $\lambda_i^l(z) = -\frac{z}{\sin\alpha}$, the highest λ_i-coordinate is $\lambda_i^h(z) = \frac{z}{\sin\alpha}$. Let $z_{i,x,y}$ denote the value of z for which the point of $\mathcal{E}(z)$ with the lowest $(x = l)$ or the highest $(x = h)$ λ_i-coordinate reaches the lower $(y = l)$ or the upper $(y = u)$ bound of the interval I_i, that is, $\lambda_i^x = \iota_y^{(i)}$. Then the value $z_{i,x,y}$ is computed as $z_{i,x,y} = \pm\iota_y^{(i)} \sin\alpha$, where the sign is plus for $x = h$ and minus for $x = l$.

Another type of the event points are the values of z where the boundary of $\mathcal{E}(z)$ coincides with a corner of the rectangle $I_1 \times I_2$. At that point one interval disappears, or a new one emerges. These event points are denoted by $z_{x,y}$, where $x, y \in \{l, u\}$ and the first index x specifies the lower (l) or upper (u) bound of the interval I_1, and the second index y specifies the bound of I_2. The $z_{x,y}$ values can be computed by plugging the corresponding interval bounds in the Equation (4.8).

Finally, the last type of the candidate event points are the distance values z_i^j, $i, j = 1, 2$, such that the λ_1-coordinates of the i-th intersection point of the boundary of $\mathcal{E}(z)$ with the line $\lambda_2 = \iota_2^l$ and of the j-th intersection point of $\partial\mathcal{E}(z)$ with the line $\lambda_2 = \iota_2^u$ are equal. At that positions the types of some intervals change. The λ_1-coordinates are computed by plugging the interval bounds of I_2 for the parameter λ_2 and solving the Equation (4.8) for λ_1.

Note that not all of the described candidate event points exist for every two segments, and not for all of the existing event points, the corresponding (λ_1, λ_2)-

points lie within the rectangle $I_1 \times I_2$. For this reason they are called candidates in the above description.

In our example, for the segments depicted in Figure 4.9(a) there exist nine event points that have the corresponding (λ_1, λ_2)-points within the rectangle $I_1 \times I_2$, and two of them coincide. The positions of the ellipse $\mathcal{E}(z)$ for each of the event points is shown in Figure 4.11.

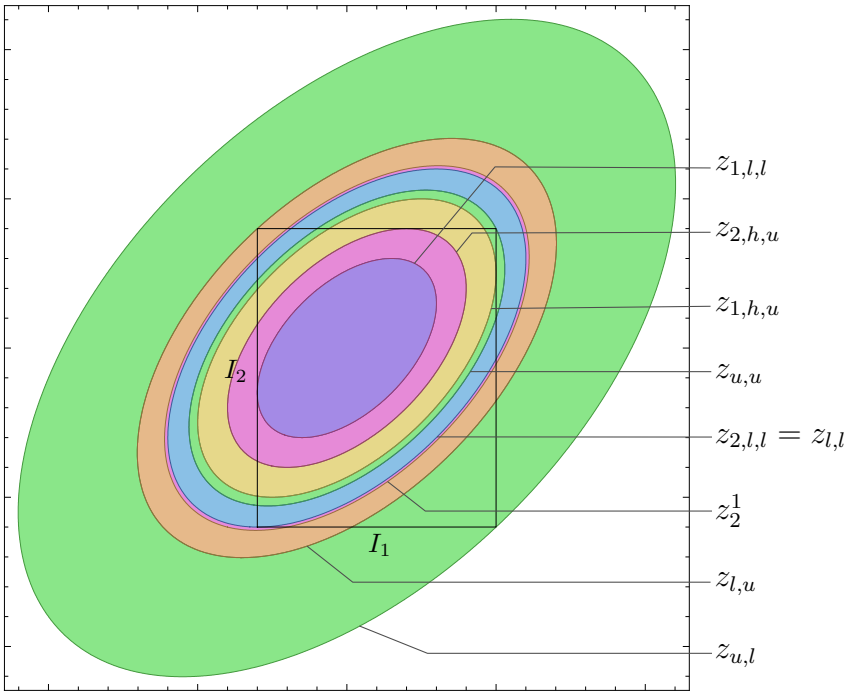


Figure 4.11: Positions of the ellipses $d^2(g_1(\lambda_1), g_2(\lambda_2)) \leq z^2$ within the rectangle defined by the intervals I_1, I_2. The contours mark the ellipse positions at the event points, and the regions of the same color denote the intervals of the distance values where the combinatorial configuration of the subdivision intervals on I_1 and I_2 does not change.

The event points subdivide the domain of the function f_{d2} into intervals for the piecewise definition. Over each interval between two event points the function f_{d2} is decomposed into a sum of the functions f'_{d2}, each defined over a subinterval of I_1. The bounds of the subintervals are determined by the bounds of I_1, ι_1^l, ι_1^u, by the lowest and highest λ_1 coordinate of the $\mathcal{E}(z)$, λ_1^l, λ_1^h, and by the intersection points of the $\partial\mathcal{E}(z)$ with the lines $\lambda_2 = \iota_2^l$ and $\lambda_2 = \iota_2^u$. In our example for $z_{2,h,u} \leq z \leq z_{1,h,u}$ the function f_{d2} is decomposed into a sum of three functions defined over four subintervals of I_1 of types ①, ③, ① and ⑤ as illustrated in Figure 4.12. Of course, the summand over the interval of type ⑤ is zero.

The complete definition of the function f_{d2} for our example in Figure 4.9(a) is listed in Table 4.3.5 and the graph of the function is depicted in Figure 4.13. The

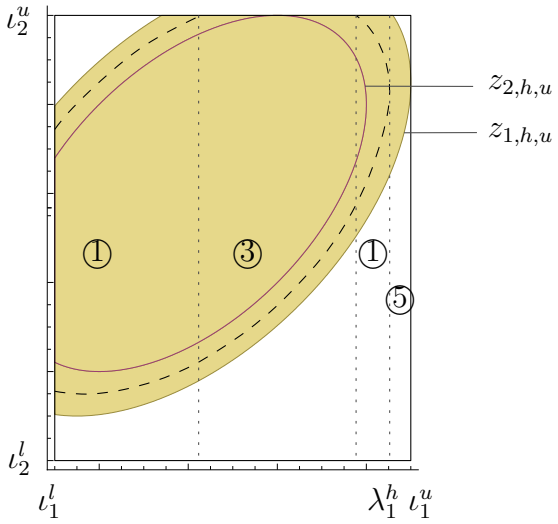


Figure 4.12: Subdivision of I_1 into subintervals for $z_{2,h,u} \leq z \leq z_{1,h,u}$.

density function for the distance distribution between two points chosen randomly on two segments (not necessarily two different segments) is defined in Equation (4.7) and is a weighted sum of f_{d2} and the density function for one segment examined in Section 4.3.1.

Since the computation of the density function for the distance distribution between two points on two arbitrary line segments involves many case distinctions, we omit the computation of the density function for the distribution of the scaling factor for shapes consisting of two line segments.

4.3.6 Distance Distribution on a Set of Line Segments

In this section we briefly consider the following experiment: Let A be a shape given as a set of line segments, $A = \{s_1, \ldots, s_n\}$. Two points $a_1, a_2 \in A$ are chosen randomly uniformly distributed with respect to length. The random variable X_d records the Euclidean distance between a_1 and a_2. The density function of the random variable X_d is a weighted sum of the distance distribution density functions for one segment, f_{d1}, examined in Section 4.3.1 and for two segments, f_{d2}, described in the previous section:

$$f_d(z) = \frac{1}{L^2}\left(\sum_{i=1}^{n} l_i^2 f_{d1}(z, s_i) + \sum_{i=1}^{n}\sum_{j=i+1}^{n} f_{d2}(z, s_i, s_j)\right) \quad ,$$

where l_i denotes the length of a segment s_i, $i = 1, \ldots, n$, and L is a total length of the segments in A. For the same reason as in the previous section we do not attempt to give a general formula for the distribution of the scaling factor generated by two sets of line segments.

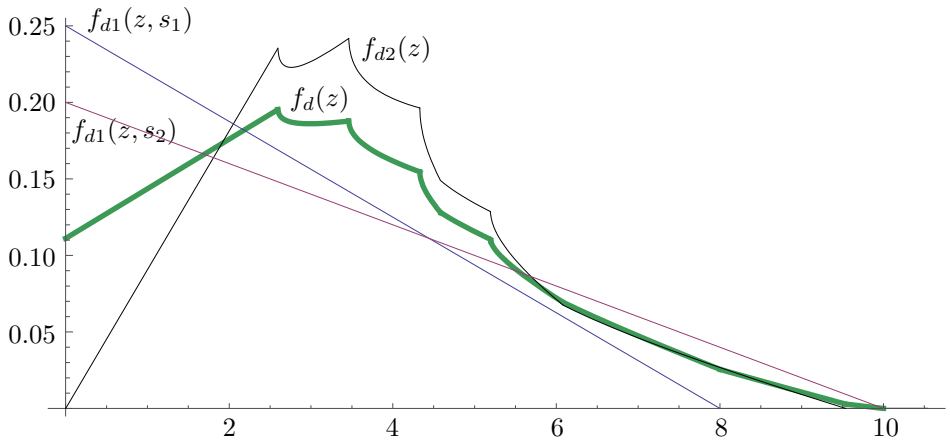


Figure 4.13: Graphs of the density functions of the distance distribution between two random points. Functions $f_{d1}(z, s_1)$ and $f_{d1}(z, s_2)$ denote the density function of the experiment where two points are selected randomly on *one* segment s_1 or s_2, respectively. Function f_{d2} is the density of distance distribution between two points such that one point is selected from segment s_1 and the other from segment s_2 (plotted with thin black line). The density function f_d of the original experiment, i.e., two points are selected independently each from one of the two segments, is a linear combination of the functions $f_{d1}(z, s_1)$, $f_{d1}(z, s_2)$ and $f_{d2}(z)$ (plotted with bold green line).

$$
f_{d2}(z) = \begin{cases}
\frac{2\pi z}{l_1 l_2 \sin\alpha} & 0 \le z \le z_{1,l,l} \\
\frac{2\pi z}{l_1 l_2 \sin\alpha} \arcsin\frac{\iota_1^l \sin\alpha}{z} & z_{1,l,l} \le z \le z_{2,h,u} \\
-\frac{z}{l_1 l_2 \sqrt{\mathcal{X}}} \Big(-2\iota_2^u \sin^2\alpha + \frac{\sqrt{\mathcal{X}}}{\sin\alpha} \Big[-\pi + 2\arcsin\frac{\iota_1^l \sin\alpha}{z} & z_{2,h,u} < z \le z_{1,h,u} \\
\quad - \arcsin\frac{\sin\alpha(\iota_2^u\cos\alpha - \sqrt{\mathcal{X}})}{z} + \arcsin\frac{\sin\alpha(\iota_2^u\cos\alpha + \sqrt{\mathcal{X}})}{z}\Big] & \\
\quad + \sqrt{z^2 - \sin^2\alpha(-\iota_2^u\cos\alpha + \sqrt{\mathcal{X}})^2} & \\
\quad + \sqrt{z^2 - \sin^2\alpha(\iota_2^u\cos\alpha + \sqrt{\mathcal{X}})^2}\Big) & \\
-\frac{z}{l_1 l_2 \sqrt{\mathcal{X}}} \Big(-2\iota_2^u \sin^2\alpha + \frac{\sqrt{\mathcal{X}}}{\sin\alpha} \Big[2\arcsin\frac{\iota_1^l \sin\alpha}{z} - 2\arcsin\frac{\iota_1^u \sin\alpha}{z} & z_{1,h,u} < z \le z_{u,u} \\
\quad - \arcsin\frac{\sin\alpha(\iota_2^u\cos\alpha - \sqrt{\mathcal{X}})}{z} + \arcsin\frac{\sin\alpha(\iota_2^u\cos\alpha + \sqrt{\mathcal{X}})}{z}\Big] & \\
\quad + \sqrt{z^2 - \sin^2\alpha(-\iota_2^u\cos\alpha + \sqrt{\mathcal{X}})^2} & \\
\quad + \sqrt{z^2 - \sin^2\alpha(\iota_2^u\cos\alpha + \sqrt{\mathcal{X}})^2}\Big) & \\
\frac{z}{l_1 l_2 \sqrt{\mathcal{X}}} \Big(-2\iota_2^u \sin^2\alpha + \cos\alpha\sqrt{\mathcal{X}} + \frac{\sqrt{\mathcal{X}}}{\sin\alpha} \Big[-2\arcsin\frac{\iota_1^l \sin\alpha}{z} & z_{u,u} < z \le z_{2,l,l} \\
\quad + \arcsin\frac{\iota_1^u \sin\alpha}{z} + \arcsin\frac{\sin\alpha(\iota_2^u\cos\alpha - \sqrt{\mathcal{X}})}{z}\Big] & \\
\quad - \sqrt{z^2 - \sin^2\alpha(-\iota_2^u\cos\alpha + \sqrt{\mathcal{X}})^2}\Big) & \\
\frac{z}{2 l_1 l_2} \Big(4\cos\alpha + \frac{1}{\sin\alpha} \Big[-2\arcsin\frac{\iota_1^l \sin\alpha}{z} + 2\arcsin\frac{\iota_1^u \sin\alpha}{z} & z_{2,l,l} < z \le z_{l,u} \\
\quad - 2\arcsin\frac{\sin\alpha(\iota_2^l\cos\alpha + \sqrt{\mathcal{Y}})}{z} + 2\arcsin\frac{\sin\alpha(\iota_2^u\cos\alpha - \sqrt{\mathcal{X}})}{z}\Big] & \\
\quad + \frac{2 l_2 \sin^2\alpha}{\sqrt{\mathcal{Y}}} & \\
\quad - \frac{2}{\sqrt{\mathcal{Y}}}\sqrt{z^2\cos^2\alpha - \iota_2^l \sin^2\alpha(\iota_2^l\cos 2\alpha + 2\cos\alpha\sqrt{\mathcal{Y}})} & \\
\quad - \frac{2}{\sqrt{\mathcal{X}}}\sqrt{z^2\cos^2\alpha - \iota_2^u \sin^2\alpha(\iota_2^u\cos 2\alpha - 2\cos\alpha\sqrt{\mathcal{X}})} & \\
\frac{z}{l_1 l_2 \sqrt{\mathcal{Y}}} \Big(-\iota_2^l \sin^2\alpha + \cos\alpha\sqrt{\mathcal{Y}} + \frac{\sqrt{\mathcal{Y}}}{\sin\alpha} \Big[\arcsin\frac{\iota_1^u \sin\alpha}{z} & z_{l,u} < z \le z_{u,l} \\
\quad - \arcsin\frac{\sin\alpha(\iota_2^l\cos\alpha + \sqrt{\mathcal{Y}})}{z}\Big] & \\
\quad - \sqrt{z^2 - \sin^2\alpha(\iota_2^l\cos\alpha + \sqrt{\mathcal{Y}})^2}\Big) & \\
0 & z < 0 \text{ or } z > z_{u,l}
\end{cases},
$$

where
$\mathcal{X} = z^2 - (\iota_2^u)^2 \sin^2\alpha$,
$\mathcal{Y} = z^2 - (\iota_2^l)^2 \sin^2\alpha$.

Table 4.1: Function $f_{d2}(z)$ for the segments depicted in Figure 4.9(a).

Note that the density distribution generated by a shape can be of interest on its own, and can be used as a characterization of a shape. Osada et al. [50] describe a method for comparing shapes based on the so called shape distributions. The main idea is to compute a certain distribution function for each shape and then measure the similarity between two shapes by the similarity of their distribution functions. The article describes different distribution functions applicable to 2D and 3D shapes. The distance distribution is reported to be most distinctive for both 2D and 3D test sets. The authors calculate the distribution functions analytically for basic shapes like line segments or a sphere and experimentally for more complex shapes. In this and the previous section we describe a method for computing the distribution function analytically for arbitrary 2D shapes represented by line segments. Of course, the analytical method is more complicated and might turn out to be more time consuming then the experimental construction. It is therefore more appropriate for applications with the main focus on exact shape analysis rather then on a large number of shape comparisons performed within a short period of time.

From our experimental evaluation of the distance distribution on the contour shapes of the ***MPEG-7 Shape B*** dataset (see Appendix A.1) we can conclude that the distance distribution alone is not a sufficient descriptor for the shape matching task. But we can confirm the observation made by Osada et al. in [50] that the distance distribution can be used to determine the general form of a shape or even the form of significant components of a shape, e.g., for mostly circular shapes (apples) the distance distribution is close to that of a circle, or for lengthy shapes with two long almost straight parallel sides (pencils) the distribution is close to that of a line segment.

Further, we experimentally constructed the density function of the scaling factor distribution for the pairs of shapes of the MPEG-7 shape B dataset, see Appendix A.2. The experiments for the shapes within the same class confirm that the scaling factor distribution for the shapes that are "similar" to a line segment, that is, long thin shapes bounded by two straight parallel line segments, e.g., a pencil, also resembles the distribution computed in the Section 4.3.2. The experimentally computed distribution of the scaling factor for such shapes has its maximum for some value close to zero. Similarly, the scaling distribution for the circular shapes, e.g., apples, resembles the distribution of the scaling factor for two circles analyzed in Section 4.3.4. We also observe that for a large group of shapes in the dataset the distribution of the scaling factor has its maximum close to zero, independent from the actual ratio of shapes. This observation is consistent with the analytically derived density function in Section 4.3 for the shapes consisting of single line segments. Therefore, we conclude that determining the scaling factor for two shapes separately by the described experiment is not meaningful in general, since even for similar shapes the maximum of the distribution tends to be close to zero.

Appendix A

Experiments on the MPEG-7 Core Experiment CE-Shape-1 Dataset

The experiments described in this section are performed on the ***MPEG-7 Core Experiment CE-Shape-1*** dataset. The Motion Picture Expert Group (MPEG), a working group of ISO/IEC (see `http://www.chiariglione.org/mpeg/`) has defined the MPEG-7 standard for description and search of audio and visual content. The data set created by the MPEG-7 committee for evaluation of shape similarity measures [13, 42] is a commonly used dataset to measure the performance of shape descriptors and shape retrieval systems. The dataset contains contour images categorized in 70 classes with 20 images in each class. The representation of shapes by polygonal curves was generated with an edge detection software.

A.1 Distance Distribution

For this experiment the shapes are normalized such that every shape has the diameter one. So the distance d between any two points of a shape is $0 \leq d \leq 1$. In one random experiment two points are selected randomly uniformly distributed from the shape and the distance between the points is recorded. The experiment is repeated 1 000 000 times, and the density function is approximated by a histogram with 100 bins.

There are some observations we would like to mention here that lead us to a conclusion that the distance distribution is not a sufficient characteristics to measure the shape similarity. But it is suitable to perform a rather coarse shape classification, e.g., to classify shape as circular, linear, elliptic, etc.

Similar shapes can have sufficiently different distribution functions. There are classes in the MPEG-7 Core Experiment CE-Shape-1 dataset for which the distance distributions of the shapes of one class differ sufficiently, some examples are given in Figure A.1. The main reason for the large variation in distance distribution

for most of these examples is that the shapes of the same class are not geometrically similar. In these cases the classification is based on the semantics and not on the geometry of the shape.

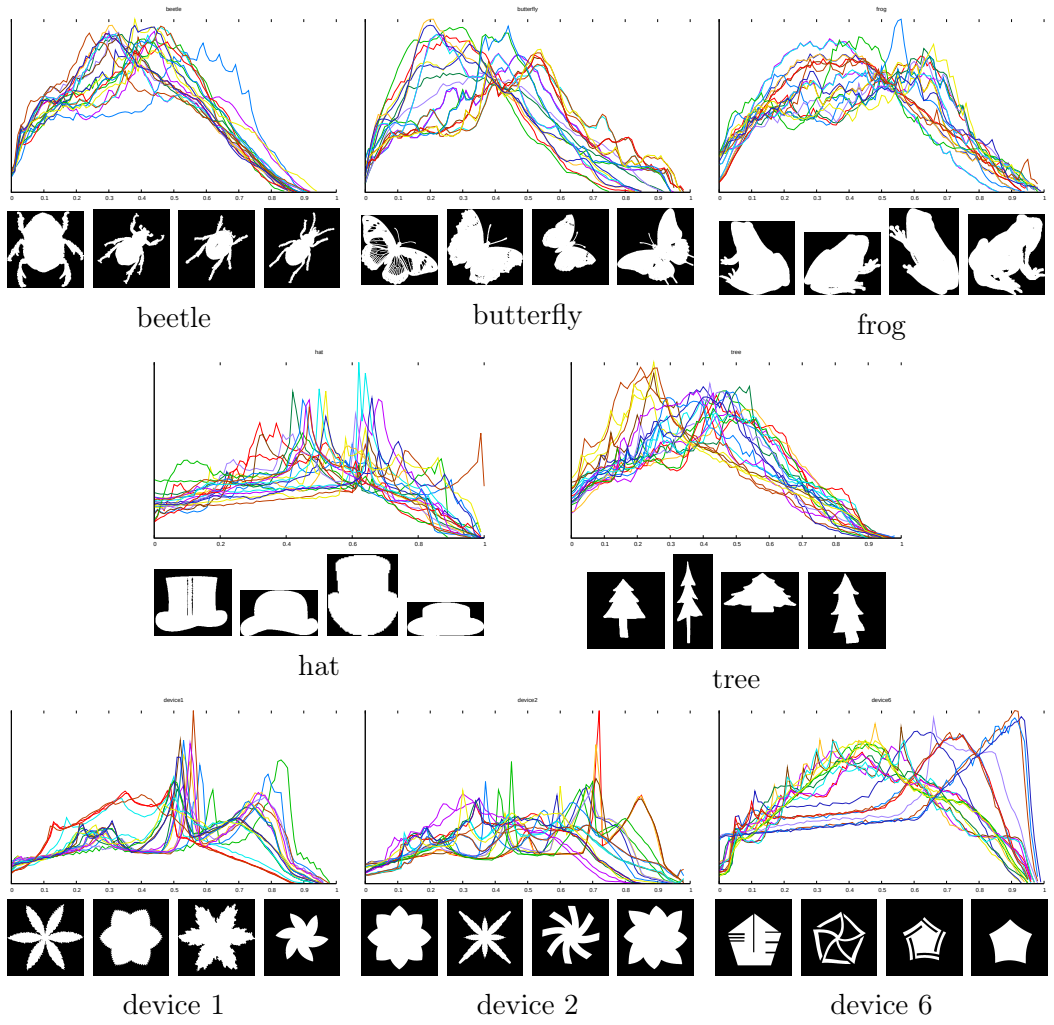


Figure A.1: Examples of classes of the MPEG-7 Core Experiment CE-Shape-1 dataset for which the distribution of the point distances does not give a good characterization of the class. Below the plots of the distance distributions for each class four examples of the images of that class are depicted.

Different shapes can have similar distributions. Figure A.2 depicts some classes with similar distance distributions. The first group shown in the first two rows consists of the classes beetle, cattle, crown, dog, fly and horse. The second group in the last row comprises camels deers and elephants. Observe that the variance within three classes depicted together in one plot (the third row in the Figure A.2) is only slightly larger than within one class. Of course one can argue that the images of

four legged animals have similar geometric properties. But it does not explain the similarity between the classes cattle and fly, where the shapes of the two classes are not geometrically similar.

Coarse shape categorization is possible. For the shapes with strong resemblance to some simple geometric figure, like line segment, circle, triangle, rectangle, etc., the distance distribution is also close to that of the corresponding geometric figure. Since we only calculated the distributions for a straight line segment and a circle analytically in this work (see Sections 4.3.1, 4.3.3) we also give examples only for the shapes with large circular or linear components.

The classes apple and device 9 depicted in Figure A.3 serve as examples for circular shapes where some irregularity or additional features disturb only a little the distance distribution of a circle. The classes with clear linear components are for example pencil, bottle, bone, and watch, see Figure A.4. Common to these shapes is that they are for the larger part slim lengthy rectangles with more (as for bone shapes) or less (as for pencil shapes) additional components. The amount of additional features is also reflected in the distance distribution function. Surprisingly, the shapes of the class brick which are neither rectangular, nor do they have long parallel lines, also show the linear character of the distance distribution. In fact, these shapes can roughly be approximated by rectangles with aspect ratio of $1:2$ to $1:3$. Additionally, for the classes like key and spoon where about half of the shape is almost a circle and about a half is a long thin rectangle we observe the presence of both components in the distance distribution function.

There are also other basic shapes that can be found in the MPEG-7 dataset and the corresponding similarities in the distribution functions, which we do not further describe here. And, as there are quite a few classes of the shapes that cannot be described by a small amount of simple geometric objects, there are correspondingly many distribution functions without clear characteristics. Examples of such shapes can be found in Figure A.2.

A.2 Scale Distribution

In this section we describe the experimental evaluation of the scaling factor distribution for the shapes in the MPEG-7 Core Experiment CE-Shape-1 dataset. We consider the distribution of the scaling factor for the pairs of shapes from the same class. For this experiment we select randomly uniformly distributed two points from each shape and record the ratio of the distances between the points. The experiment is repeated 500 000 times, and the density function is approximated by a histogram with bin width 0.1. The graphs of the density function in the figures of this section are restricted to the maximum scaling factor value 5. In all scale distribution graphs of this section we plot the results of comparisons of all 20 shapes of one class to the first figure of that class.

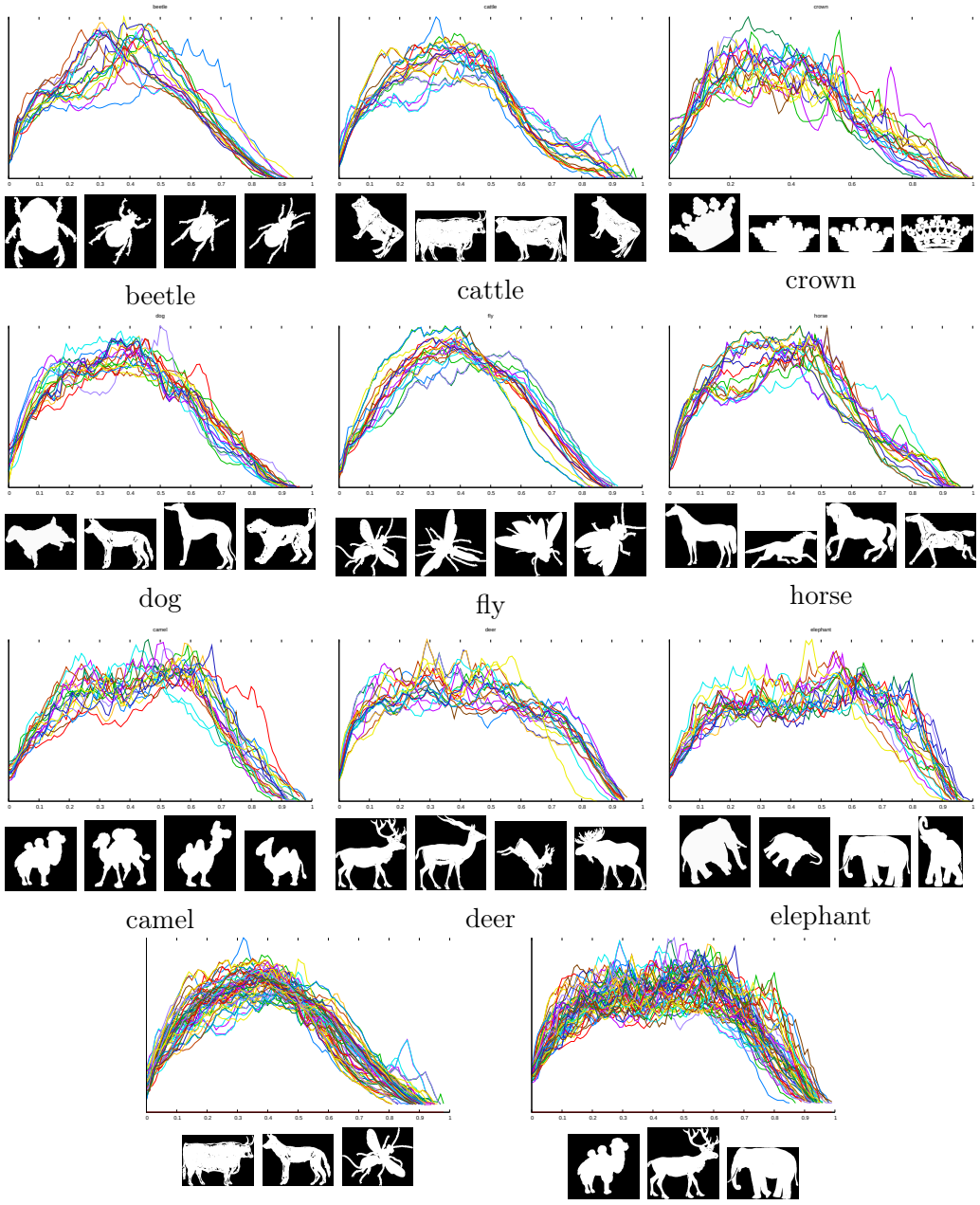


Figure A.2: Examples of the MPEG-7 Core Experiment CE-Shape-1 dataset classes for which the distribution of the point distances is similar for shapes that belong to different classes. The left figure in the last row depicts the distance distribution curves for the classes cattle, dog, and fly, and the right figure – for the classes camel, deer, and elephant.

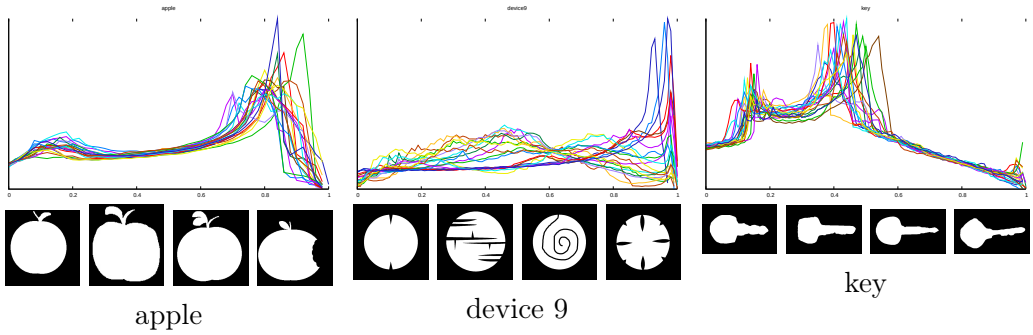


Figure A.3: Density function of the distance distribution. Examples of the MPEG-7 Core Experiment CE-Shape-1 dataset classes with circular shape: apple and device 9, and partially circular shape: key.

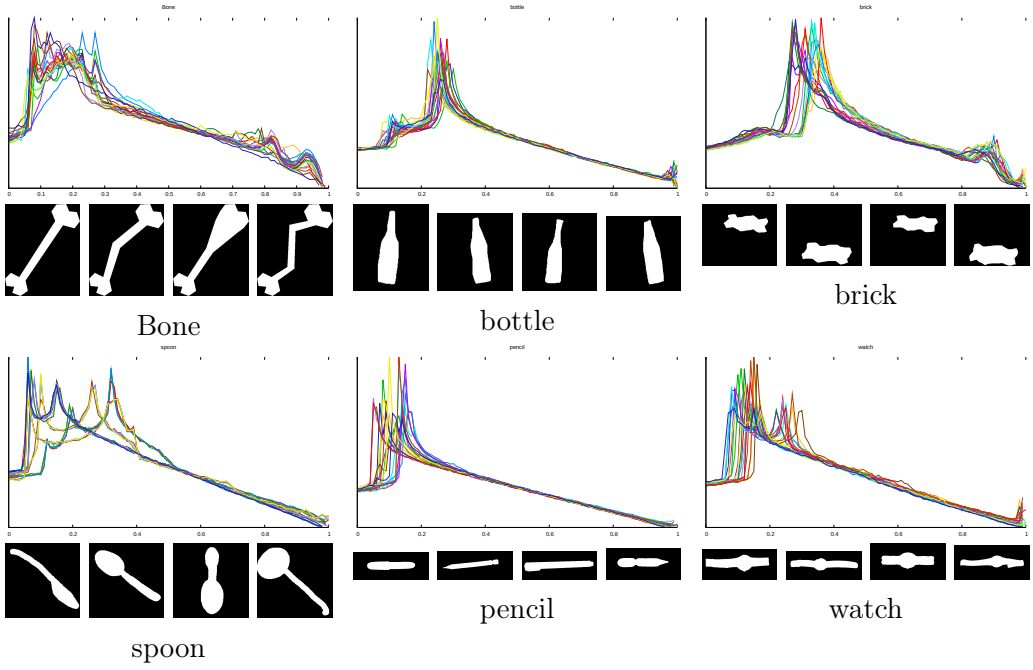


Figure A.4: Density function of the distance distribution. Examples of the MPEG-7 Core Experiment CE-Shape-1 dataset classes with large linear component.

It is not surprising that for the classes of clearly circular shapes, such as apple and device 9, the distribution of the scaling factor also corresponds to that of two circles, see Figure A.5. For such shapes the distribution of the scaling factor has a clear maximum, which corresponds to the factor that scales the second shape to the size of the first.

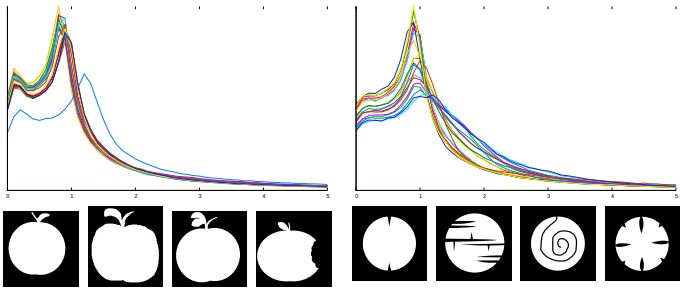


Figure A.5: Classes of the MPEG-7 Core Experiment CE-Shape-1 dataset with circular shape: apple and device 9. The distribution of the scaling factor corresponds to that of the circles.

It is also not surprising that the shapes that have a long thin rectangle as their major component also have a scale distribution that is similar to the scale distribution for two line segments, see Figure A.6. The maximum of the scale distribution for such shapes is although not exactly at zero but is very close to zero.

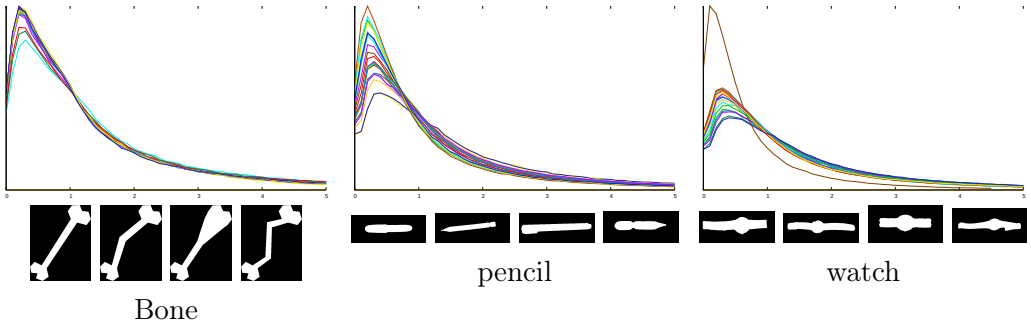


Figure A.6: For the classes with a large linear component the distribution of the scaling factor corresponds closely to the distribution of the scaling factor for two line segments.

Interestingly, the scaling factor distribution for the shapes with large circular and large linear components, like key and spoon, that had a mixed structure in the distance distribution function, behaves mostly like the scaling distribution for the line segments as depicted in Figure A.7. That means that the maximum of the scaling factor distribution is close to zero for these shapes.

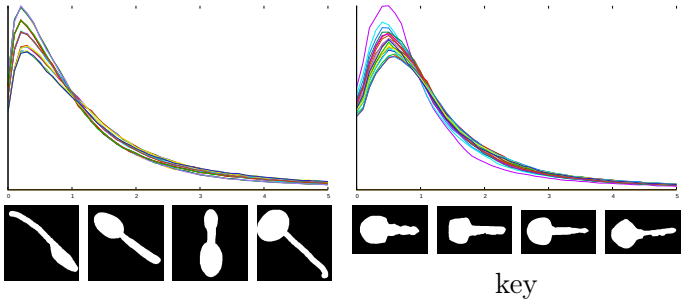


Figure A.7: Classes with a "mixed" structure: the shapes have a large circular as well as a large linear part. The maximum of the scaling factor distribution tends to be close to zero.

Furthermore, it turns out that for 52.8% (37 out of 70) of the shape classes of the MPEG-7 Core Experiment CE-Shape-1 dataset the distribution of the scaling factor has its maximum close to zero for almost all pairwise comparisons within the class. Therefore, the experiment in this form is not suitable for computing the appropriate scaling factor for two shapes as a first step of the matching under similarity transformations.

With the modification described in Section 4.3.2, where the scale distribution function $f_s(z)$ is multiplied by the scaling factor z, that is $f(z) = z \cdot f_{s(z)}$ in order to penalize the downscaling of the shapes, we find that the maxima of the function $f(z)$ for the shapes of the same class actually correspond to the scaling factor that transforms one shape to be approximately of the size of the other. We only state that the determined scaling factor is *approximately* the best one since it is not always possible to determine exactly the scaling factor that maximizes the similarity measure defined in Chapter 3. The shapes of the dataset are not exactly congruent and in general there exist multiple transformations maximizing that measure. Figure A.8 shows the modified scaling distribution for the classes depicted in Figures A.5 to A.7.

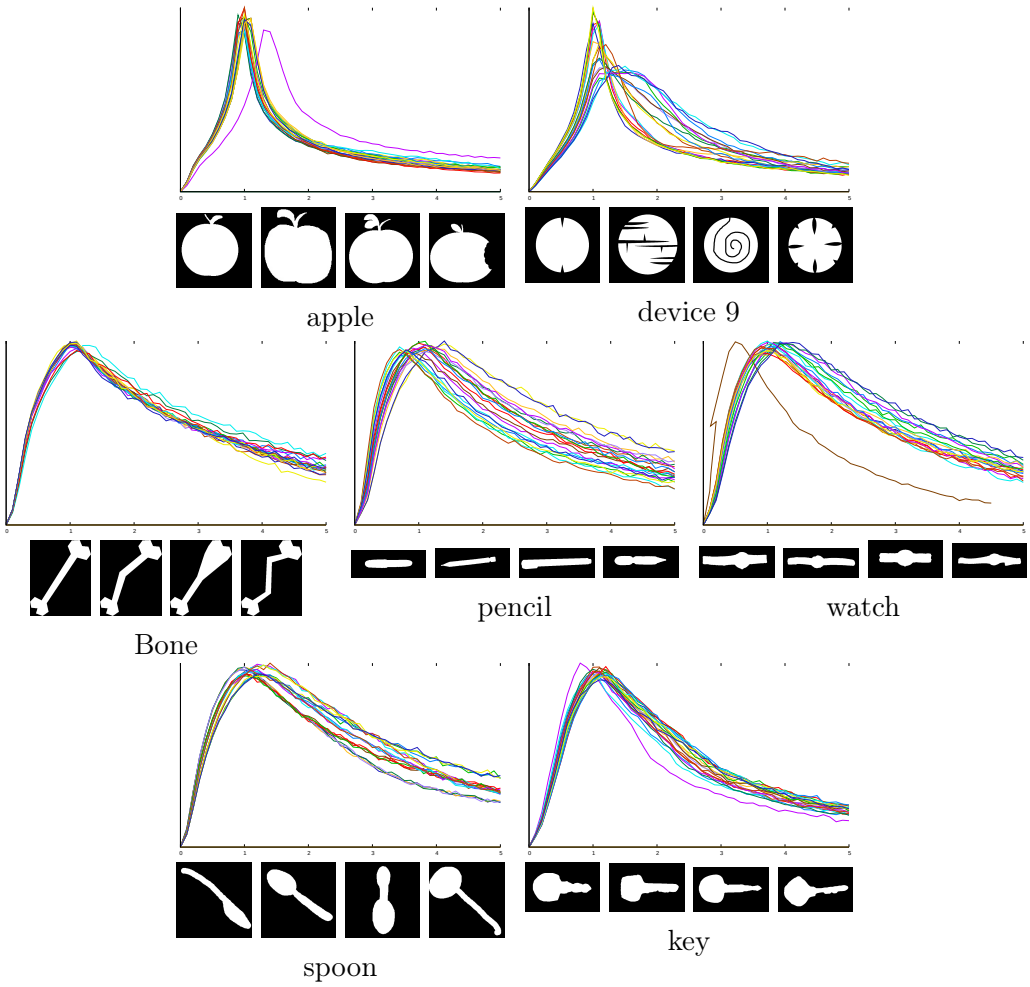


Figure A.8: Modified scaling distribution function $f(z) = z \cdot f_s(z)$ for the classes depicted in Figures A.5 to A.7.

Bibliography

[1] A. S. Aguado, E. Montiel, and M. S. Nixon. Invariant characterisation of the Hough transform for pose estimation of arbitrary shapes. *Pattern Recognition*, 35:1083–1097, 2002.

[2] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5:75–91, 1995.

[3] H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In J.-R. Sack and J. Urrutia, editors, *Handbook of computational geometry*. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 1999.

[4] H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, 13:251–265, 1995.

[5] H. Alt, G. Rote, C. Wenk, and A. Efrat. Matching planar maps. *J. of Algorithms*, pages 262–283, 2003.

[6] H. Alt, L. Scharf, and S. Scholz. Probabilistic matching and resemblance evaluation of shapes in trademark images. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 533–540, New York, NY, USA, 2007. ACM Press.

[7] H. Alt, L. Scharf, and D. Schymura. Probabilistic matching of polygons. In *Proceedings of the 24th European Workshop on Computational Geometry (EuroCG)*, Nancy, France, March 2008.

[8] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(3):209–216, March 1991.

[9] B. Aronov and S. Har-Peled. On approximating the depth and related problems. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 886–894, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[10] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954.

[11] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 168–173, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-785-5. doi: http://doi.acm.org/10.1145/237814.237857.

[12] S. Basu, R. Pollack, and M.-F. Roy. On computing a set of points meeting every cell defined by a family of polynomials on a variety. *Journal of Complexity*, 13 (1):28–37, 1997.

[13] M. Bober, J. D. Kim, H. K. Kim, Y. S. Kim, W.-Y. Kim, and K. Muller. Summary of the results in shape descriptor core experiment. ISO/IEC JTC1/SC29/WG11/MPEG99/M4869, 1999.

[14] P. Brault and H. Mounier. Automated transformation-invariant shape recognition through wavelet multiresolution. In A. F. Laine, M. A. Unser, and A. Aldroubi, editors, *Wavelets: Applications in Signal and Image Processing IX*, volume 4478, pages 434–443. SPIE, 2001.

[15] T. Bui, G. Chen, and L. Feng. An orthonormal-shell-fourier descriptor for rapid matching of patterns in image database. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(8):1213–1229, 2001.

[16] O. Cheong, A. Efrat, and S. Har-Peled. On finding a guard that sees most and a shop that sells most. In *Proc. 15th ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 1091–1100, 2004.

[17] G. Chuang and C. Kuo. Wavelet descriptor of planar curves: Theory and applications. *Ieee Transactions On Image Processing*, 5(1):56–70, 1996.

[18] M. Cui, J. Femiani, J. Hu, P. Wonka, and A. Razdan. Curve matching for open 2d curves. *Pattern Recognition Letters*, 30(1):1–10, 2009.

[19] A. Driemel. Multiscale curvature matching for smooth polylines. Master's thesis, Freie Universität Berlin, 2009.

[20] R. S. Drysdale, G. Rote, and A. Sturm. Approximation of an open polygonal curve with a minimum number of circular arcs and biarcs. *Comput. Geom. Theory Appl.*, 41(1-2):31–47, 2008.

[21] A. Dumitrescu and G. Rote. On the Fréchet distance of a set of curves. In *Proceedings of the 16th Canadian Conference on Computational Geometry*, pages 162–165, Montreal, 2004.

[22] J. P. Eakins, J. M. Boardman, and M. E. Graham. Similarity retrieval of trademark images. *IEEE MultiMedia*, 5(2):53–63, 1998.

[23] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, 15(2):341–363, 1986.

[24] A. Efrat, Q. Fan, and S. Venkatasubramanian. Curve matching, time warping, and light fields: New algorithms for computing similarity between curves. *J. Math. Imaging Vis.*, 27(3):203–216, 2007.

[25] J. Feldman and M. Singh. Information along contours and object boundaries. *Psychological Review*, 112:243–252, 2005.

[26] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[27] A. Folkers and H. Samet. Content-based image retrieval using fourier descriptors on a logo database. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3*, page 30521. IEEE Computer Society, 2002.

[28] M. Frenkel and R. Basri. Curve matching using the fast marching method. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 35–51, 2003.

[29] I. Gel'fand, M. Graev, and N. Vilenkin. *Generalized Functions. Volume 5, Integral Geometry and Representation Theory.* Academic Press, 1966.

[30] M. v. Ginkel, C. L. Hendriks, and L. van Vliet. A short introduction to the Radon and Hough transforms and how they relate to each other. Technical Report QI-2004-01, Imaging Science and Technology Department, Delft University of Technology, The Netherlands, 2004.

[31] M. v. Ginkel, C. Luengo Hendriks, P. Verbeek, and L. van Vliet. The generalized radon transform: Sampling, accuracy and memory considerations. *Pattern Recognition*, 38(12):2494–2505, December 2005.

[32] G. Grimmett and D. Welsh. *Probability, an Introduction.* Oxford University Press, 1986.

[33] M. Hagedoorn. *Pattern matching using siilarity measures.* PhD thesis, Utrecht University, the Netherlands, 2000.

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

[35] M. Hagedoorn, M. Overmars, and R. Veltkamp. A new visibility partition for affine pattern matching. In *Proc. Discrete Geometry for Computer Imagery conference, DGCI 2000*, pages 358–370, Berlin, 2000. Springer-Verlag.

[36] S. Hermann and R. Klette. A comparative study on 2D curvature estimators. In *Proceedings of the International Conference on Computing: Theory and Applications (ICCTA'07)*, pages 584–589, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[37] D. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.

[38] W. Kaplan. *Advanced Calculus*. Addison-Wesley, 4th edition, 1992.

[39] I. Kunttu, L. Lepisto, J. Rauhamaa, and A. Visa. Multiscale fourier descriptor for shape-based image retrieval. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, pages 765–768. IEEE Computer Society, 2004.

[40] R. Lakaemper and M. Sobel. Correspondences between parts of shapes with particle filters. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[41] L. J. Latecki and R. C. Veltkamp. Properties and performances of shape similarity measures. In *Proceedings of International Conference on Data Science and Classification (IFCS)*, 2006.

[42] L. J. Latecki, R. Lakämper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 424–429, 2000.

[43] L. J. Latecki, R. Lakämper, and M. Sobel. Polygonal approximation of point sets. *Combinatorial Image Analysis*, pages 159–173, 2006.

[44] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3D geometry. *ACM Trans. Graph.*, 25(3):560–568, 2006.

[45] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[46] M. Mohajer. A random algorihm to determine resemblance between two 3d-surfaces. Master's thesis, Freie Universität Berlin, Department of Computer Science, January 2008.

[47] F. P. Nava and A. F. Martel. Planar shape representation based on multiwavelets. In *EUPSICO 2000 : European signal processing conference*, Signal processing X : theories and applications, pages 1751–1753, 2000.

[48] C. F. Olson. Efficient pose clustering using a randomized algorithm. *Int. J. Comput. Vision*, 23(2):131–147, 1997.

[49] C. F. Olson. *Fast Object Recognition by Selectively Examining Hypotheses.* PhD thesis, University of California at Berkeley, 1994.

[50] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, 2002.

[51] J. Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. In *Berichte Sächsische Akademie der Wissenschaften, Leipzig*, volume 69, pages 262–277. Mathematisch-Physikalische Klasse, 1917.

[52] B. E. Rogowitz and R. Voss. Shape perception and low-dimension fractal boundary contours. In *Proc. SPIE, Human Vision and Electronic Imaging: Models, Methods, and Applications*, volume 1249, pages 387–394, Oct. 1990.

[53] T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(1):116–125, 2003.

[54] B. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, New York, 1986.

[55] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40:361–387, 1987.

[56] M. Tanase, R. C. Veltkamp, and H. Haverkort. Multiple polyline to polygon matching. In *Proceedings of the 16th Annual Symposium on Algorithms and Computation (ISAAC 2005), LNCS 3827*, pages 60–70, 2005.

[57] P. Toft. *The Radon Transform - Theory and Implementation.* PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, 1996.

[58] Z. Tu, S. Zheng, and A. Yuille. Shape matching and registration by data-driven em. *Comput. Vis. Image Underst.*, 109(3):290–304, 2008.

[59] H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science and Engineering*, 04(4):10–21, 1997.

Lebenslauf

Der Lebenslauf ist in der Online-Version aus Gründen des Datenschutzes nicht enthalten