

Visual Shape Similarity and Retrieval of Figurative Images

Dissertation zur Erlangung des Doktorgrades

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

von

Sven Scholz

2010

Betreuer: Prof. Dr. Helmut Alt

Gutachter: Prof. Dr. Helmut Alt
Prof. Dr. Remco C. Veltkamp

Datum der Disputation: 18.05.2011

The work presented in Chapter 2 was financially supported by AKTOR Knowledge Technology NV, Project TCiMage.

The work presented in Chapters 3 and 4 was financially supported by the European Union under contract No. IST-511572-2, Project Perceptually-Relevant Retrieval of Figurative Images (PROFI).

Preface

“THE question ‘What makes things seem alike or seem different?’ is one so fundamental to psychology that very few psychologists have been naïve enough to ask it.”

In 1950 Fred Attneave precluded his article *Dimensions of Similarity* [16] with this provocative statement. Much has changed since that time and a lot of progress has been achieved by psychologists investigating the essence of perceived similarity. However, in computer science there is often still a large gap between the modeling of similarity on the one side and the real-world behavior of what should have been modeled on the other side. The author of the present work does not claim that he is able to close, or even narrow this gap but wants to bring it a little bit more to awareness. In order to do so, the basic properties of perceived similarity and the main misconceptions are discussed in the first chapter, before the actual topic, namely comparing figurative images based on the depicted shapes, is treated in Chapters 2 to 4.

• • ♦ ♦ ♦ ♦ • •

I would like to thank all those who accompanied me on my way, who gave me support and made all this possible. . .

Contents

Introduction	13
1. Fundamentals	17
1.1. Images	17
1.1.1. Generic Images	17
1.1.2. Shapes	19
1.1.3. Figurative Images	21
1.2. Perception	24
1.2.1. Basics	24
1.2.2. Perception of Shapes	26
1.2.3. Prototypicality and Saliency	28
1.2.4. Completion of Good Form	29
1.2.5. Superiority over Automated Detection	30
1.3. Similarity	32
1.3.1. Generic Definitions	32
1.3.2. Possible Properties of Measures	33
1.3.3. Perceived Similarity	36
1.3.4. Some Examples of Measures and Models	46

Contents

1.4. Retrieval	53
1.4.1. Basic Concepts	53
1.4.2. Retrieval Performance	55
1.4.3. Trademark Image Retrieval	56
1.5. Significance of Heuristic Approaches	60
1.5.1. Basics	60
1.5.2. Heuristics in the Context of Similarity Estimation	61
1.5.3. No Free Lunch	61
1.6. Data	62
1.6.1. Requirements	62
1.6.2. Employed Data Sets	63
2. Extraction of Shapes	67
2.1. Basics	68
2.2. Vectorization of Figurative Images	70
2.2.1. Related Work	70
2.2.2. Discretization of Colors	70
2.2.3. Boundary Detection	76
2.3. Merging of Small Shapes	77
2.3.1. Textured Regions	77
2.3.2. Broken Lines	81
2.4. Simplification	90
2.4.1. Polyline Simplification	90
2.4.2. Deletion of Irrelevant Data	107
2.5. Grouping	110

3. Estimation of Shape-Similarity	113
3.1. Basics	113
3.2. Mapping	116
3.2.1. Related Work	116
3.2.2. Idea of the Proposed Approach	119
3.2.3. Sampling	121
3.2.4. Computing Transformations	124
3.2.5. Weighting Transformations	129
3.2.6. Clustering Transformations	130
3.2.7. Analysis	136
3.3. Similarity Estimation	138
3.3.1. Related Work	138
3.3.2. Idea of the Proposed Approach	142
3.3.3. Definition of the Similarity Measure	143
3.3.4. Computation of the Similarity Measure	145
3.3.5. Analysis	146
3.3.6. Properties	146
3.4. Evaluation	148
4. A Framework for Automated Trademark Image Retrieval	153
4.1. Related Work	154
4.2. Extraction of Shapes	160
4.2.1. Vectorization	160
4.2.2. Merging	161
4.2.3. Simplification	162
4.3. Similarity Estimation based on Mapping	162

Contents

4.4. Similarity Evaluation based on Image Primitives	164
4.4.1. Idea of the Proposed Approach	164
4.4.2. Basic Perceptual Units and Relationships	165
4.4.3. Extraction of Figures	166
4.4.4. Comparison of Images	168
4.4.5. Experimental Results	169
4.5. Conclusion and Future Work	170
A. Experiments on the UK Trademarks Set	173
B. Specification of Parameter Values	187
B.1. Extraction of Shapes	187
B.2. Similarity Estimation based on Mapping	196
B.3. Similarity Estimation based on Image Primitives	197
Bibliography	201
Summary	225
Zusammenfassung	227

Introduction

THE free online dictionary *Wictionary* [240] defines *similarity* as the “*closeness of appearance to something else*”. Assuming that the closer two objects’ appearances are, the more properties they share, it becomes obvious that perceiving visual similarity is a very fundamental and important ability: It is an evolutionary advantage to be able to derive the properties of an object, e. g., whether an animal is dangerous or not, by comparing its appearance with the appearances of known objects.

The fields *object recognition* (identifying a known object), *classification* (identifying an unknown object as a member of a known class of objects), and *similarity estimation* are closely related as the connection between *similarity* and *confusability* suggests (see section 1.3.3.1). However, there is a crucial difference: Normally, the question whether a scene contains a specific object (or whether a scene contains an instance of a class of objects, respectively) has a definite answer which is *yes* or *no*—even if finding the correct answer may be hard to achieve. The answer to the question whether two objects are perceived similar on the other hand is of gradual nature and depends on the observer. Things that have sufficiently many properties in common will be perceived similar, but which properties do suffice and how does the number and the selection of common properties affect the magnitude of perceived similarity? This topic has been subject to ambitious research but is not yet fully understood.

In many domains of object recognition and classification, human judgement is superior to current automatic systems¹, in some domains deficiencies of human judgement became apparent and other criteria were desired², but there are also domains where determining perceived similarity is by definition the main objective. One of these domains is trademark image retrieval.

¹ For this reason some approaches for automatic object recognition have been inspired by nature see, e. g., [75, 155]

² For example for identifying persons based on passport photographs, visual inspection is replaced by automatic comparison of biometric features.

Introduction

Trademark images are invented to increase the recognition value of a company, its goods or services. In order to establish the wanted linkage between a trademark image and the owning company, the image has to be distinctive, non-confusable. Due to the growing number of marks—in some databases it has reached millions, and hundreds of new marks are added every day—services, such as finding trademark images confusingly similar to a given one, or making sure that no such similar ones exist, become more and more important and at the same time, offering such services becomes more and more challenging.

The most prominent features of many trademark images are the depicted shapes whereas color and texture are only used to form these shapes. Therefore, trademark image retrieval would benefit a lot from the ability to extract and compare shapes automatically. Unfortunately, shape seems to be a feature hard to handle and many generic image retrieval systems using shape information produce results that do not conform to perceived shape similarity [219]. However, limiting the domain to figurative images such as trademark images makes the detection of shapes easier and therefore facilitates the development of retrieval systems which produce reasonable results.

In Chapter 1 basic concepts used in connection with content-based image retrieval are introduced and an overview of the state of knowledge on human perception and perceived similarity is given. Chapter 2 deals with the extraction of shapes, which serves as a basis for the comparison of figurative images. Several issues that have to be considered in this context are discussed, an overview of the approaches that have been proposed for solving the problems is given, and new algorithms that especially allow for the demands of shape extraction from figurative images are presented. Chapter 3 deals with the comparison of shapes, which is the main focus of this work. An approach for measuring the similarity is presented, which is based on techniques originally used for object recognition instead of similarity estimation. Finally, in Chapter 4 all the presented approaches are combined in a framework for automated trademark image retrieval. In order to demonstrate their usability in practice, all the presented algorithms have been implemented and tested on various sets of figurative images and real-world trademark images.

Fundamentals

In this chapter the basic concepts used in the context of content-based image retrieval are introduced. It gives an overview of the mathematical and psychological basics of similarity estimation and outlines the special needs of trademark image retrieval.

1.1. Images

Nowadays, images are ubiquitous. Along with their usage in, e. g., advertisements, newspapers, on television, in the internet, in computer games etc. there is a huge number of different forms of appearance: digitally, printed, painted; as photographs, artwork, drawings, illustrations, pictograms. . .

What do they all have in common, and how do they differ?

1.1.1. Generic Images

Formally, a two-dimensional *image* can be seen as a function that maps every point of a (for simplicity rectangular) region $R \subset \mathbb{R} \times \mathbb{R}$ to an element of some color space C . Depending on this color space an image is called *color image* in the most general case, *gray level image* for $C = [0, 1]$ (spuriously often referred to as black-and-white image), or *bi-level black-and-white image* for $C = \{0, 1\}$; in both cases 0 standing for *black* and 1 standing for *white*.

1. Fundamentals

This abstract definition allows to code arbitrary information in an image, but in the given context the visual effects of images on human viewers are of interest. Therefore, an image informally is associated with a *content* (what it depicts—which itself may be something abstract), with one or several *representations* (how it is coded or materialized), and with its *perception* (what is evoked in the viewer’s mind). The perception of an image clearly also depends on the actual representation since the information has to be made available to the viewer’s eye somehow. A criterion for the quality of a given representation, therefore, might be the accuracy with which viewers would be able to reconstruct the original image based on this representation.¹

In information technology generic images are mostly represented as *raster graphics* where the function mapping to some color space is only defined for discrete points, so called *pixels*, of a rectangular grid $G \subset \mathbb{N} \times \mathbb{N}$. Depending on the device used to visualize such raster graphics, a pixel p with coordinates (x, y) can be interpreted, e. g., as the square $[x, x + 1[\times [y, y + 1[$ or as a disc with center (x, y) and some fixed radius r . Based on the first interpretation, two pixels will be called *edge-neighboring* if the corresponding squares’ boundaries share an edge, or *corner-neighboring* if the corresponding squares’ boundaries share a point. Common formats for storing such raster graphics are among others

- PNG (Portable Network Graphics) which is a format using lossless compression of raster graphics with up to 2^{48} colors (cf. [124]).
- JPEG (Joint Photographic Experts Group) [125] which is actually not a file format but a description of methods for the compression of raster graphics. One important technique is based on partitioning the grid into blocks of 8×8 pixels. After the color information of each block has undergone a discrete cosine transform, it gets quantized which—depending on the compression rate—causes the typical jpeg-artefacts (an example can be seen in Figure 2.2 on page 69).

In order to represent an image by a raster graphic it has to be discretized. This should be done in a way that the resulting representation (in conjunction with an appropriate output device, such as a screen or a printer) has a maximum quality in the sense outlined above. Some techniques for discretization have gained broad acceptance. For example in bi-level black-and-white raster graphics a one-dimensional object l , for instance a straight line segment, is

¹ Please recall that the visual effects on viewers are considered here. In this sense a paper with the written words “black square” on it is not regarded as a good representation for an image depicting a black square although a viewer with knowledge about the English language and geometric concepts might perfectly reconstruct the original image.

typically represented by a chain of connected pixels such that no four of them are pairwise edge-neighboring and the distances of the pixels to l are small. A quite simple algorithm for this problem was invented by Bresenham [34].² Depending on the resolution, the inevitable stair-case behavior of these chains of pixels may be irritating. In gray level images so called *anti-aliasing* techniques can be applied to achieve a smoothed representation with higher quality. The color of a pixel is interpolated between black and white depending on the pixel's distance to the object or depending on the ratio of its area that is covered by the object³. A very common algorithm for the anti-aliased rasterization of lines or edges was presented by Wu [247]. Some examples of rasterized straight line segments are shown in Figure 2.2 on page 69.

Another challenge is the conversion of a gray-level image to a bi-level black-and-white image. The illusion of having different gray levels can be created by blending black and white pixels⁴ using different densities. One of the first and still commonly applied solutions to this problem is the Floyd-Steinberg dithering [87] (an example can be seen in Figure 1.1). Moreover, the ideas of this algorithm can also be utilized when the number of different colors in a raster graphic is to be reduced.

Unfortunately, rasterization and some of the techniques that are utilized to facilitate better—with respect to perception—representations of images make the automated analysis of the content more difficult.

1.1.2. Shapes

Shape plays a crucial role in human object classification and identification [174], and it is also regarded as being one of the predominant features determining the perceived similarity of images [231]. The term *shape*, however, is used in a variety of meanings.

In [99] a shape is formally defined as a subset of a Euclidean space. Sometimes this subset is also required to be compact. According to this definition a two-dimensional shape s can be described by its characteristic function $f_s: \mathbb{R}^2 \rightarrow \{0, 1\}$.

² Bresenham's algorithm was originally intended for drawing straight line segments with digital plotters, but the idea applies to raster graphics as well. Moreover, it has also been generalized to other classes of curves.

³ In practice, the area ratios are often estimated using multiple discrete sample points per pixel (supersampling).

⁴ This technique is usually applied to raster graphics, but using an appropriate tiling it might of course also be applied to images with a continuous domain.

1. Fundamentals

In geometry, on the other hand, a shape is defined as an equivalence class, meaning that two sets have the same shape if one can be transformed to the other by a combination of translations, rotations, and uniform scalings.⁵

Actually, the common meaning of *shape* is less precisely defined: The term is used for the “*appearance of something, especially its outline*” [240]. As an object is not restricted to have a single color, in a depiction the object’s shape does not necessarily equal a region of homogeneous color. This is why the self-evident equality of the definition of a bi-level black-and-white image and the characteristic function of a shape does not correspond to human perception (see Figure 1.1 for an example).

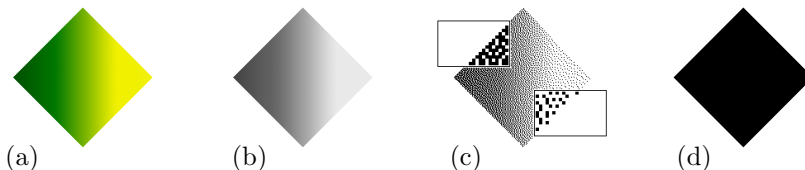


Figure 1.1. Image and shape:

(a) color image depicting a square, (b) the image converted to gray level, (c) the image converted to bi-level black-and-white using Floyd-Steinberg dithering⁶ (and two scaled up details), (d) the perceived shape.

Two things have to be differentiated between: firstly, the definition of shape, and secondly, the questions in which ways such shapes are depicted in images, how visual stimuli lead to the perception of shapes, and how shapes can be extracted from images.

In the present work the term shape is used in a rather restrictive sense according to the following definitions:

- A *line shape* is a simple plane curve. Analogously, a *polygonal line shape* is a simple piecewise linear plane curve (polyline).

⁵ This point of view will be used for shape similarity estimation in chapter 3.

⁶ Images (a) and (b) as *you* actually see them, might—depending on the used output device—also be dithered, but at a higher resolution than image (c) so hopefully, it will not be as apparent.

- A *simple region shape* is a compact two-dimensional region bounded by a simple closed curve. Analogously, a *simple polygonal region shape*, or *simple polygon* for short, is a compact two-dimensional region bounded by a simple closed polyline.
- A *region shape* may be seen as a simple region shape (eventually) with holes. Formally, it can be defined by induction:
 - Every simple region shape is a region shape.
 - Let S be a region shape with boundary $\partial(S)$ and let H be a simple region shape with boundary $\partial(H)$ such that $\partial(S) \cap \partial(H) = \emptyset$, then $S \setminus (H \setminus \partial(H))$ is a region shape.

Polygonal region shapes may be defined analogously.

In the following, whenever the type of shape is not explicitly stated, *shape* is meant to refer to *simple region shapes*. Since a specific polyline P can be characterized by the sequence (p_1, \dots, p_n) of points (the vertices) where the linear pieces (the edges) are connected, the notation $P = (p_1, \dots, p_n)$ will be used here.

1.1.3. Figurative Images

Unlike in figurative art, in the present context the term *figurative* does not refer to the question *what* is depicted in an image, but *how* it is depicted. Figurative images differ from natural images like, e. g., photographs by the fact that the content is artificially produced, stylized, and that shape is emphasized. Most figurative images are of low complexity: only a few colors are used, boundaries between different colors are clear cut, and the number of depicted objects is small. On the other hand, since they are artificially produced, figurative images may be designed using stylistic methods such as, e. g.,

- depicting a shape only by its outline,
- depicting a shape by a textured region,⁷
- hatching (some special form of texture),
- depicting a shape implicitly.

⁷ The region is filled with some pattern. In the given context texture differs from dithering by the fact, that structures are explicitly perceptible. They just do not lead to the perception of distinct shapes, but are seen as feature of shapes they form.

1. Fundamentals

Exemplary instantiations of these stylistic methods can be seen in Figures 1.2 and 1.3. Some real-world trademark images using some of these stylistic methods can be seen in Figure 1.4.

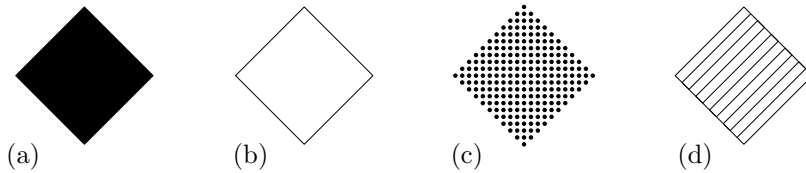


Figure 1.2. Depiction of shapes:
(a) square depicted by a filled region, (b) square depicted by its outline, (c) square depicted by a textured region (small filled circles), (d) square depicted by its outline and a hatched region.

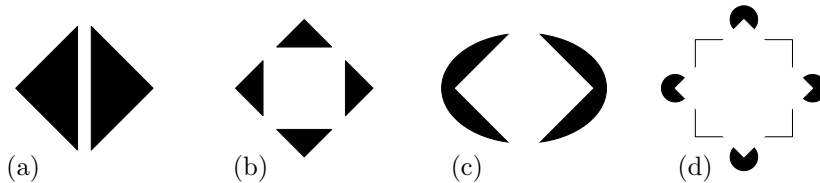


Figure 1.3. Implicit shapes:
(a) square formed by two triangles that are separated by a thin stripe, (b) the same square formed by four triangles that are separated by a region, (c) square formed by the space inbetween two other shapes, (d) implicit square formed by seemingly occluded shapes (adaption of the *Kanizsa triangle*).

Figurative images may appear, e. g., as icons (pictograms), trademark images, coats of arms, clip-art images, etc. The purpose of pictograms is to convey a simple, straightforward message. As to do so, they often contain well known symbols or stylized depictions of real-world objects. Trademark images often also depict totally abstract geometrical shapes (see Figures 1.4 and 1.16 for some examples).

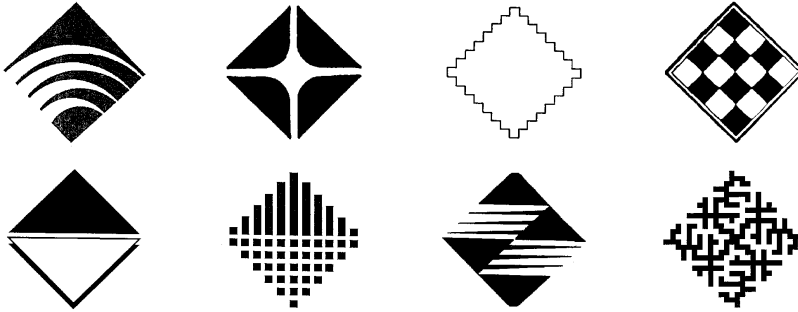


Figure 1.4. Stylistic methods in trademark images:
eight real-world trademark images essentially depicting the same square in different ways.

The databases of trademark registries still contain lots of old bi-level black-and-white images that have been scanned; some of them containing a huge amount of so called *salt-and-pepper noise* (erroneous white and black pixels) or artefacts from the scanning (see Figure 1.5 (a) for an example). Nowadays trademark images are often made available in digital form, which means that automated image retrieval could benefit from straight and high quality images. However, these images are often of low resolution and contain compression artefacts (see Figure 1.5 (b) for an example).

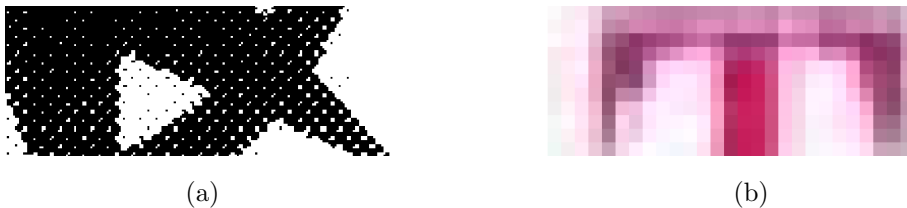


Figure 1.5. Poor quality images:
details from images of poor quality due to (a) scanning the image, (b) low resolution and compression artefacts.

1.2. Perception

Goldmeier [96] stated that similarity is a relation not between the physical stimuli but between the perceptions of these stimuli.⁸ Mach [158] gave the simple example that two triangles with side lengths a, b, c and $a + m, b + m, c + m$, respectively, are not necessarily perceived similar although there is a very simple relation between the side lengths. On the other hand, patterns that are maximally dissimilar in some mathematical models may be perceived as virtually the same.⁹ A serious consideration of visual similarity, therefore, cannot ignore the factor perception.

1.2.1. Basics

For simple stimuli the dependence of the perceived intensity p on the physical magnitude s is described by the *psychophysical function* $p = f(s)$. The smallest difference in the physical magnitude s that is detectable by a human is called *just noticeable difference (jnd)* Δs . Based on Weber's [237] observation, that in a limited range the quotient $W(s) = \frac{\Delta s}{s}$ is almost constant, Fechner [79] inferred that the perceived intensity is logarithmically depending on the physical magnitude: $p = k \cdot \log(s/s_0)$ for some constant k and a minimum perceivable magnitude s_0 . Stevens [211], on the other hand, claimed that the psychophysical function can be described by a power law: $p = k \cdot s^\alpha$ for some constant k and an exponent α depending on the type of stimuli at hand. In recent years the psychophysical function has also been analyzed by means of mathematical constraints (see, e. g., [157]).

Stevens already mentioned that his power law would not hold for the complete range of magnitudes [211], however, it also seems questionable whether such constrained classes of functions can be appropriate for all different kinds of stimuli. Moreover, also contextual factors might influence the perception of a stimulus which is demonstrated by a very simple experiment (as carried out, e. g., in high school): A subject lays one hand in a basin filled with cold water, the other one in a basin filled with warm water. After putting both hands in a third basin filled with water of medium temperature—actually not knowing that it is a single basin—virtually every subject states that on one side the water has higher temperature than on the other side. The impact of context on the perception of abstract stimuli was confirmed, e. g., in an experiment where subjects were to categorize the area of squares. The categories subjects chose changed depending on the frequency of small and large squares [212].

⁸ “Die Ähnlichkeit ist eine Beziehung, die nicht zwischen den Reizkomplexen, sondern zwischen deren anschaulichen Bildern besteht.”

⁹ For example random patterns with same parameters under some transformational models (see Section 1.3.4 page 52 and Figure 1.12).

The visual stimuli evoked by images may be composed of many simple stimuli and therefore may be very complex. The input we get by our senses has to be ‘compressed’ somehow—otherwise we would have to store something like 10 000 terabytes for visual stimuli alone [168]. Attneave [17] mentions as an example for such a compression or abstraction, that two random patterns with same parameters would be perceived as being virtually the same (see Figure 1.12 on page 52). He states that “*Any sort of physical invariance whatsoever constitutes a source of redundancy for an organism capable of abstracting the invariance and utilizing it appropriately, but we actually know very little about the limits of the human perceptual machinery with respect to such abilities.*”

According to *feature integration theory*, perception can be subdivided into different stages. Simple visual features like color, orientation, and spatial frequency are registered in an early, preattentive stage and are combined in a later stage to form the perception of objects [217]. Moreover there is empirical evidence, that even objects have cognitive representations on several levels of abstraction: Among others, a viewpoint-dependent representation of the object as currently seen, called *object token*, and an object-centered representation from which the object’s appearance from other viewpoints can be predicted, called *structural description* [218]. Such a structural description is, of course, not uniquely determined by the stimuli, so the question arises, which structural description(s) brain chooses.

One tendency seems to be obvious: The simpler a structural description is, the more likely it will be chosen. A popular example that encourages this assumption is the Kanizsa triangle (an adaption of it can be seen in Figure 1.3): The image explicitly depicts 3 disks each of which has a circular sector cut off, and 3 angles. Interpreting this pattern as originating from 3 complete discs and a triangle, all partially occluded by an additional triangle, reduces the complexity of the structural description; the number of objects is reduced from 6 to 5 and all the shapes have more symmetries.

This tendency, called the *minimum principle*, was also confirmed experimentally, e.g., by using patterns of straight line segments that could either be interpreted as edges of two-dimensional shapes in the plane, or as projection of the edges of a three-dimensional object. Sequences of patterns were generated, such that for a complete sequence the three-dimensional interpretations were the same object just seen from different viewpoints, and the corresponding two-dimensional interpretations were of different complexity (in terms of number of angles and continuous lines). With increasing complexity of the two-dimensional interpretations the tendency to perceive the pattern as a three-dimensional object also increased [111]. In general, simple patterns which have a high degree of internal redundancy, called *good gestalt* [17], are easier to recognize; they are, e.g., associated with shorter classification reaction times (see [30] for an overview).

1. Fundamentals

However, external redundancy (conformance with known patterns or concepts) may also affect perception. The so called *Helmholtz principle* [107] states that a stimuli configuration leads to the perception of an object which normally would cause this stimuli configuration.¹⁰ A conclusion drawn from this principle is, that we perceive what is most likely [111].^{11,12} The adaptation of the perceptual system to stimuli that frequently occur (imprinting) has been confirmed in many studies (see [98] for an overview), but also higher level regularities may influence the further processing of patterns: For example sequences of words are easier to remember if they form a correct sentence than if they are ordered arbitrarily [164].

1.2.2. Perception of Shapes

The perception of a distinct shape may be evoked by very different depictions. An object might be discernable because it has a homogeneous color such that its projection appears to be a region with no abrupt changes in the color gradient on a background of different color. However, an object might also be discernable because it has a homogeneous texture such that its projection appears to be a region with no abrupt changes in the texture gradient on a background of different texture (see [94] for a detailed discussion). In both cases the information about the object is concentrated along the contour of its projection (especially at those points where the direction of the contour changes most rapidly) [17]. Objects are already quite recognizable from only these contours [195] and therefore, simplified line drawings serve as good (concerning reaction times and error rates in recognition tasks) as full color photographs [29].

Shapes are normally rather perceived as objects that may have additional attributes such as color or texture, however, also contour parts discernable in an image—even if not continuous and not bounding closed regions—may lead

¹⁰ *“Die allgemeine Regel, durch welche sich die Gesichtsvorstellungen bestimmen die wir bilden, wenn unter irgend welchen Bedingungen oder mit Hilfe von optischen Instrumenten ein Eindruck auf das Auge gemacht worden ist, ist die, dass wir stets solche Objecte als im Gesichtsfelde vorhanden uns vorstellen, wie sie vorhanden sein müssten, um unter den gewöhnlichen normalen Bedingungen des Gebrauchs unserer Augen denselben Eindruck auf den Nervenapparat hervorzubringen.”*

¹¹ Another interpretation of the Helmholtz principle is sometimes formulated in the following way: “We perceive what is most *unlikely*,” meaning that the patterns that reach consciousness are the ones that could not come into being by chance. For a detailed discussion of this point of view see, e. g., [61].

¹² The observation that simple structural descriptions are preferred, might also be brought into accordance with the Helmholtz principle by assuming simplicity to increase the likeliness.

to the perception of a shape. Different parts, possibly depicted in different ways, are somehow grouped together to form shapes. In the beginning of the nineteenth century a number of principles were formulated that predict certain perceptions to be more likely than others. These principles are subsumed under the name *Gestalt theory* and are based on the holistic point of view, that—in simple terms—the whole is more than the sum of its parts.

Among psychologists, Gestalt theory has been criticized for only describing, but not explaining the observed phenomena. In the given context, however, knowing the principles may help to improve automated image analysis and therefore, some of them are listed in the following (see [35] for a more detailed overview).

proximity Things that are close together are grouped together.

similarity Things that are similar are grouped together.

good continuation Things (for example contours) that can be glued together smoothly without abrupt changes in direction, are grouped together.

closure Groupings that produce closed contours are preferred.

symmetry Symmetric contours, rather than others, lead to the perception of shape (figure).

These principles describe very general tendencies. For getting adequate—in terms of estimating the perceived similarity of images—representations, however, it would be important to know how the resulting patterns are further processed. For example a group of discrete elements that are very small compared to the whole are not perceived as individuals, but contribute to the perception of texture (material) rather than shape (form). This effect was observed in experiments for some instances with 7 up to 9 identical elements and for most instances with more than 9 identical elements [96].

1. Fundamentals

1.2.3. Prototypicality and Salience

According to *prototype theory*, the objects we perceive are classified and assigned to categories, so that nonidentical stimuli can be treated as equivalent. Categories are formed such that they are maximally differentiable from each other—the number of attributes shared by members of the same category is maximized, the number of attributes shared by members of different categories is minimized [195]. The categories are highly structured internally and do not have well defined boundaries. There may be a *prototype* or *clearest case* which represents the core meaning, ‘surrounded’ by other category members with decreasing similarity to that core meaning [193].

In experiments, such prototypes, e. g., perfect circles and perfect squares, were better recognized than distorted versions. Prototypes are also used to describe other category members by verbalizing the basic form plus the variation, e. g., “*It’s a square with a hole in it.*” [193]. If long-term memory works in a similar way, fading of memory could even cause a tendency towards the prototype when the additional information about variations gets lost.

The same experiments revealed that also the recognition across categories differs. Circles were learned with fewer errors than squares, and squares with fewer errors than triangles [193]. This might be explained by the goodness of the gestalts as mentioned above. It is argued that when two figures are roughly equivalent with respect to goodness of gestalt, the more complex figure is likely to be more salient, but that a good gestalt is likely to be more salient than a bad gestalt although the latter is generally more complex [222].

In connection with the salience of parts of figurative images the concept of *frames* is very important. In [70] the *border* of a symbol is defined as “*a boundary that completely surrounds a symbol, with no parts of the symbol extending beyond that boundary. Various elements of the symbol may touch the border.*” Squares, rectangles, rhombi, triangles, circles, and ellipses (ovals) are listed as possible shapes of such borders. A frame might be characterized in a very similar way, as a convex shape (preferably from the list above) containing the actual content of the image. Even though frames normally are good gestalts, their salience is supposed to be very low, meaning that they do only have a very small impact on the perceived similarity of figurative images (see Figure 4.2 on page 165 for an example). The distinction between more important content and less important frames (outline/background) is also made in [245] and the validity is underpinned by the following observation: In an experiment described in [112], subjects were instructed to redraw their perceptions of figurative images presented to them. For four of the five (out of 50) published images that have a frame, most of the subjects completely ignored the frame as if it was not a part of the actual image at all.

1.2.4. Completion of Good Form

Since objects have to be recognized also when the sensed information is incomplete, perception is supported by some ‘repairing’ mechanisms: *“To complete unfinished forms, people will ignore gaps, filling in the missing parts with a familiar pattern in order to complete the form.”* [85] (cited in [45]). Adapting the sensed stimuli to well known patterns, again leads to simpler structural descriptions of what is perceived.

Experiments by Warren [235] confirm, that under certain circumstances data that is not compatible with higher level patterns is repaired before it reaches consciousness. Mumford [169] reports on an experiment with recorded speech where a single phoneme was replaced by noise such that a word became irreognizable. However, subjects did not perceive the defect but believed they had heard the one phoneme which made the sentence semantically consistent:¹³

actual sound	perceived words
the ★eel is on the shoe	the h -eel is on the shoe
the ★eel is on the car	the wh -eel is on the car
the ★eel is on the table	the m -eel is on the table
the ★eel is on the orange	the p -eel is on the orange

A similar mechanism was explored in visual perception: The missing visual information of the region corresponding to the blind spot is somehow completed using the information from the surrounding areas. Gaps in straight lines and even more complex figures are filled extending the borders, whereas the corner of a square is not filled in when falling into the blind spot. This suggests that the mechanism intervenes at an early stage of perception, because it is not based on the viewers expectation and on learned concepts. However, patients that have a sharply localized damage in their visual cortex resulting in a small island of blindness in the visual field (called scotoma) tend to see the corner of the square after some seconds [188].

Another repairing mechanism that goes beyond filling in missing information was also observed in experiments with scotoma patients. When they were presented with images of two vertical, non collinear line segments such that the lower endpoint of the upper segment and the upper endpoint of the lower

¹³ The table is taken from an article by Mumford [169]. He refers to an article by Warren [235] which does actually not report on that special experiment. Independently, the result is also mentioned by Barsalou [21]. He refers to another article by Warren et al. [236]—unfortunately, that one also does not report on the experiment. The author of the present work, however, tends to believe that the observations really have been made.

1. Fundamentals

segment both fell into the blind region, they reported that both segments seemed to move towards each other to form a single segment [189]. This replacement of two line segments by a single line again supports the assumption, that simple structural descriptions are preferred over more complex ones.

1.2.5. Superiority over Automated Detection

By means as described above, our perception enables us to recognize objects even under obfuscated conditions. An extreme example is shown in Figure 1.6: In this image there is no obvious cue that helps to distinguish between figure and ground, but after a little while we manage to see the Dalmatian dog that is depicted, and then it remains obvious. For automatic detection, however, such images will remain hard cases in the near future.



Figure 1.6. Concealed shape: The depicted patterns are seemingly meaningless at first sight, but once the shape of the Dalmatian dog is recognized, it seems to be obviously there (Photograph by Ronald C. James published in [101]).

Figurative images might even be tailored to perception. Knowledge of the human visual system is, e.g., used to improve the presentation of scientific data [105]. But even without such knowledge, by trial and error, figurative images may be created that employ the mechanisms of the visual system to evoke a specific perception. It is not necessary to know why or how the perception is evoked, the fact that it is evoked suffices. For this reason, images might be suited to perception rather than to automated detection which even increases the difficulties in emulating visual perception.

The superiority of perception to automatic detection may also be utilized. Offering nonpaid services in the internet entails the risk of massive automated (mis-)usage. If the service shall be publicly available, but usage shall be restricted to natural persons, so called *CAPTCHA*¹⁴ systems can be employed: The prospective user is confronted with a task that can relatively easy be solved by a natural person but is believed to be hard to be solved (within reasonable time bounds) automatically. Typically this task is to read and retype a random text which is depicted in a distorted and noisy way (see Figure 1.7 for some examples).

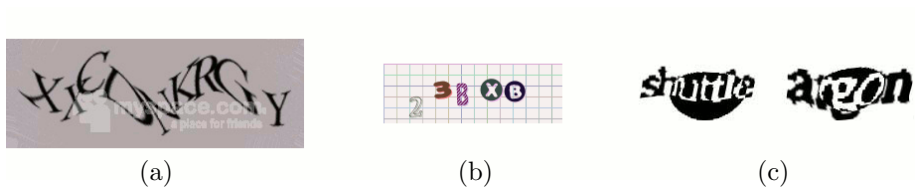


Figure 1.7. CAPTCHAs taken from social networking websites:
 (a) MySpace 2008, (b) StudiVZ 2008, (c) Facebook 2010.

¹⁴ CAPTCHA stands for ‘Completely Automated Public Turing test to tell Computers and Humans Apart’.

1.3. Similarity

Similarity is a very fundamental concept. Therefore, using the name in a formal way still often evokes connections to properties that simply need not be fulfilled. Perceived similarity, and similarity with respect to some mathematical model are two different things and the values may (and in many cases do) differ considerably.

1.3.1. Generic Definitions

In many publications the terms ‘similarity’ and ‘distance’ are used synonymously, which seems not to be appropriate for two reasons: firstly, according to linguistic usage similarity is a *closeness* and thus the value of similarity of two things is high if they (nearly) equal each other and it gets lower the more they differ, and secondly, the term distance is formally used for functions that fulfill the *metric* properties (as defined below). For the sake of clarification, the following two very basic definitions are used here:

A similarity measure σ on a set S is a real valued function $\sigma: S \times S \rightarrow \mathbb{R}$.

A dissimilarity measure δ on a set S is a real valued function $\delta: S \times S \rightarrow \mathbb{R}$.

In these generic definitions the words ‘similarity’ and ‘dissimilarity’ have no real meaning—based on linguistic usage, only some properties such a measure *should* have are indicated. The actual meaning is given by the definition of a concrete measure of similarity (or dissimilarity) and it should not be confused with the task at hand, for which this measure seems to be appropriate.

Surprisingly, this distinction is often left out of consideration which leads to questionable conclusions. In [203], e. g., the fact that an algorithm is employed to *identify* persons based on passport photographs (which was traditionally done by visual inspection), is interpreted as evidence that this algorithm computes the *perceived similarity* of the depicted persons.

Naming two real valued functions ‘similarity measure’ formally does not imply common properties. Nevertheless, perceived similarity and similarity according to arbitrary mathematical models are often mixed up, so whenever there is the risk of ambiguity, the formulation “objects a and b are similar” needs to be supplemented with the specification of the concrete similarity measure that is referred to.

1.3.2. Possible Properties of Measures

Range of Values, Conversion A similarity measure is typically supposed to assign the maximum value 1 to a pair of identical objects, and a dissimilarity measure is typically supposed to assign the minimum value 0 to a pair of identical objects.¹⁵

A measure of similarity (or dissimilarity respectively) is called *normalized* if its co-domain is the unit interval $[0, 1]$.

Given a normalized similarity measure σ , a canonical way to define a corresponding normalized dissimilarity measure δ could surely be $\delta(s_1, s_2) := 1 - \sigma(s_1, s_2)$ and vice versa—at least the function that transforms one value into the other should be strictly monotone decreasing. A detailed discussion on how to derive a similarity measure from a dissimilarity measure can be found in [203]. In general, however, for a given similarity measure and a given dissimilarity measure such a function transforming the values need not exist, because similarity and dissimilarity are defined on pairs of objects.

Metric Properties A real valued function $d: S \times S \rightarrow \mathbb{R}$ on a set S is called *metric* or *distance measure* if the following conditions are fulfilled for all $s_1, s_2, s_3 \in S$ (terminology according to [54]¹⁶):

1. non-negativity
 $d(s_1, s_2) \geq 0$
2. small self-distance
 $d(s_1, s_1) = 0$
3. isolation
 $s_1 \neq s_2$ implies $d(s_1, s_2) > 0$
4. symmetry
 $d(s_1, s_2) = d(s_2, s_1)$
5. triangle inequality
 $d(s_1, s_3) \leq d(s_1, s_2) + d(s_2, s_3)$

¹⁵ For some measures used in connection with perceived similarity this is surprisingly not the case—see Section 1.3.3.3 for details.

¹⁶ In this definition 2. and 3. imply 1. but mostly the two conditions are merged to: ‘ $d(s_1, s_1) = 0$ if and only if $s_1 = s_2$ ’. Sometimes 1. and 2. are merged to ‘*minimality*: $d(s_1, s_2) \geq d(s_1, s_1) = 0$ ’ which does not imply 3.

1. Fundamentals

A real valued function that fulfills 1.–4. but does not fulfill the triangle inequality, is sometimes called *semimetric*, a real valued function that fulfills 1.–2. and 4.–5. but does not fulfill the isolation criterion is sometimes called *pseudometric*—the common usage of these notations, however, is not consistent.

The metric axioms can easily be satisfied, e. g., by letting $\delta(a, b) := 0$ if $a = b$ and $\delta(a, b) := 1$ if $a \neq b$, but such a measure is normally not useful in practice as it gives no information about the degree of differentness. Useful measures of dissimilarity, on the other hand, are often desired also to fulfill the metric properties, because that facilitates the use of indexing structures (see [48] for an overview).

Invariance Mostly it is not differentiated between two possible ways of defining the term *invariance* but here, both definitions are listed separately: Given a class \mathcal{T} of transformations $t: S \rightarrow S$ and a real valued function $f: S \times S \rightarrow \mathbb{R}$

- f is called *invariant with respect to synchronous transformation* under the class \mathcal{T} if for all $t \in \mathcal{T}$ and for all $s_1, s_2 \in S$ the transformation t does not change the value of f when applied to both elements:

$$f(t(s_1), t(s_2)) = f(s_1, s_2).$$
- f is called *invariant with respect to asynchronous transformation* under the class \mathcal{T} if for all $t_1, t_2 \in \mathcal{T}$ and for all $s_1, s_2 \in S$ the transformations t_1 and t_2 do not change the value of f when applied to the elements:

$$f(t_1(s_1), t_2(s_2)) = f(s_1, s_2).$$

Given a measure of similarity σ (or dissimilarity δ respectively) that is not invariant with respect to asynchronous transformation under a class of transformations \mathcal{T} , a measure of similarity $\sigma^{\mathcal{T}}$ (or dissimilarity $\delta^{\mathcal{T}}$ respectively) that is in fact invariant under the class \mathcal{T} may be derived by taking the optimum over all pairs of transformations in \mathcal{T} :

$$\begin{aligned}\sigma^{\mathcal{T}}(s_1, s_2) &= \max_{t_1, t_2 \in \mathcal{T}} \left\{ \sigma(t_1(s_1), t_2(s_2)) \right\}, \\ \delta^{\mathcal{T}}(s_1, s_2) &= \min_{t_1, t_2 \in \mathcal{T}} \left\{ \delta(t_1(s_1), t_2(s_2)) \right\}.\end{aligned}$$

Robustness In [229] four kinds of *robustness* for distance measures on patterns were defined. For arbitrary measures (which do not necessarily fulfill the triangle inequality) of dissimilarity and similarity on sets of shapes these definitions have to be slightly adapted: Let S be a collection of sets of shapes (in the following a shape set $s \in S$ is interpreted as the union of its elements), and let $f: S \times S \rightarrow \mathbb{R}$ be a real valued function.

- Deformation robustness is the property that small deformations of shapes only result in small changes of the result. Let T be the maximal group of homeomorphisms under which S is closed. f is called *deformation robust* (with respect to the second argument)¹⁷ if for each $s_1, s_2 \in S$ and $\varepsilon > 0$ there is a $\mu > 0$, such that $|f(s_1, t(s_2)) - f(s_1, s_2)| < \varepsilon$ for all $t \in T$ satisfying $\|x - t(x)\| < \mu$ for all $x \in \partial(s_2)$.
- Blur robustness is the property that additions close to the boundary of shapes do not cause discontinuities. f is called *blur robust* (with respect to the second argument)¹⁷ if for each $s_1, s_2 \in S$ and $\varepsilon > 0$ an open neighborhood U of $\partial(s_2)$ exists, such that $|f(s_1, s'_2) - f(s_1, s_2)| < \varepsilon$ for all $s'_2 \in S$ satisfying $s'_2 \setminus U = s_2 \setminus U$ and $\partial(s_2) \subseteq \partial(s'_2)$.
- Crack robustness is the property that changes within a small neighborhood of a (non singleton) boundary point only result in small changes of the result, regardless whether the connectedness is preserved or not. Given a shape set s , a singleton point $q \in s$ is a point such that an open neighborhood V of q exists with $V \cap s = \{q\}$. Let $\zeta(s)$ denote the set of singleton points of s . f is called *crack robust* (with respect to the second argument)¹⁷ if for each $s_1, s_2 \in S$, each $p \in \partial(s_2) \setminus \zeta(s_2)$, and $\varepsilon > 0$ an open neighborhood U of p exists, such that $|f(s_1, s'_2) - f(s_1, s_2)| < \varepsilon$ for all $s'_2 \in S$ satisfying $s'_2 \setminus U = s_2 \setminus U$.
- Noise robustness is the property that changes within small regions in the complement of the shapes only result in small changes of the result. f is called *noise robust* (with respect to the second argument)¹⁷ if for each $s_1, s_2 \in S$, for each finite point set $P \subseteq \mathbb{R}^2 \setminus \partial(s_2)$, and $\varepsilon > 0$ an open neighborhood U of P exists, such that $|f(s_1, s'_2) - f(s_1, s_2)| < \varepsilon$ for all $s'_2 \in S$ satisfying $s'_2 \setminus U = s_2 \setminus U$.

Distributivity A dissimilarity measure $\delta: S \times S \rightarrow \mathbb{R}$ on a set S is called *distributive* (with respect to the second argument)¹⁷ if for all $s_1, p, q \in S$ such that $p \cap q = \emptyset$, the value for s_1 and the union of the parts does not exceed the sum of values for s_1 and the parts: $\delta(s_1, p \dot{\cup} q) \leq \delta(s_1, p) + \delta(s_1, q)$. [230]

¹⁷ The definition with respect to the first argument is analogous.

1. Fundamentals

Sensitivity Sensitivity is the property that changes of parts where two items equal each other increase the value of a dissimilarity measure. Let $\delta: S \times S \rightarrow \mathbb{R}$ be a dissimilarity measure on a set S . δ is called *sensitive* (with respect to the second argument)¹⁷ if for all $s_1, s_2, s'_2 \in S$ such that $s_1 \cap U = s_2 \cap U$, $s'_2 \setminus U = s_2 \setminus U$, and $s'_2 \cap U \neq s_2 \cap U$ for some U , $\delta(s_1, s'_2) > \delta(s_1, s_2)$. [230]

Monotonicity Monotonicity means that for 2 non-counteracting changes the minimum dissimilarity caused by one of them alone has to be smaller than the dissimilarity caused by the combined change. Let $\delta: S \times S \rightarrow \mathbb{R}$ be a dissimilarity measure (that fullfills the symmetry property) on a set S . δ is called *strictly monotone* if for all $s_1, s_2, s_3 \in S$ such that $s_1 \subset s_2 \subset s_3$, $\delta(s_1, s_2) < \delta(s_1, s_3)$ or $\delta(s_2, s_3) < \delta(s_1, s_3)$. [230]

1.3.3. Perceived Similarity

The way stimuli are perceived and how similarities are judged is not entirely understood, but there has been extensive research and some of the results and interpretations might help to improve the quality of automatically computed ratings of similarity.

1.3.3.1. Measurements

Perceived similarity σ_p cannot be measured directly, but there are some quantities that are used to explore its nature. Most of the experimental approaches can be grouped together in three classes:

judged similarity Subjects make statements about their judgements on similarity. Either

- quantifying by giving ratings of judged similarity σ_j of two given stimuli on a scale, or
- comparative by choosing from a given set of stimuli either the one that is judged most similar to a reference stimulus or the pair of stimuli that is judged most similar.

The ratings of judged similarity σ_j are typically assumed to agree only ordinally with perceived similarity. It is commonly assumed, however, that $\sigma_p = f(\sigma_j)$ with f being a monotonic function [15].

confusability In so called recognition experiments, subjects have to recognize or classify stimuli. Properties of perceived similarity are then derived, e. g.,

- from reaction times or
- from the relative frequency $\varrho_{s,r}$ of the event that presenting stimulus s results in response r . The diagonal entries $\varrho_{i,i}$ correspond to correct recognition, off-diagonal entries $\varrho_{i,j}$ with $i \neq j$ correspond to erroneous recognition.

Subjects tend to generally prefer some answers to others—a phenomenon known as biased choice. It is assumed, however, that in the absence of such response bias, perceived similarity is proportional to the empirical probability of confusion (erroneous recognition) [15].¹⁸ The diagonal entries $\varrho_{i,i}$ are regarded as indicators for the perceived similarity of stimuli to themselves. These values may differ from stimulus to stimulus. In [138] it is argued that (self) similarity also depends on the density of stimuli and therefore, on the stimulus domain.¹⁹

transfer An organism that has been trained to respond in some predictable manner to a particular stimulus-situation is presented with a modified stimulus-situation. The relative frequency of the original response under the modified conditions gives an indication of perceived similarity [16].

The different natures of the experiments are also reflected in the data that is generated. Therefore the derived results are not always consistent. For example digits $\mathbb{5}$ and $\mathbb{9}$ have been rated more similar than $\mathbb{5}$ and $\mathbb{8}$ although the latter were more often confused in recognition tasks [92].

Ignoring the problems mentioned above, in the following there will be made no difference between perceived similarity and the measures it is derived from, because the authors of the original papers mostly don't.

¹⁸ Please note, that the motivation behind the present work is to automatically identify trademark images that are similar in the sense that they have the potential for getting confused.

¹⁹ The phenomenon might be examined in a very simple imaginary experiment: Given the three stimuli equilateral triangle, circle, and circle with dot in the center; since the circles with and without dot are hard to distinguish, their diagonal entries will be much lower than for the triangle. On the other hand, given the three stimuli equilateral triangle, equilateral triangle with dot in the center, and circle, the diagonal entries for the triangles will be much lower than for the circle. The relative numbers of correct recognitions of triangle and circle are supposed to be very different depending on the domain.

1. Fundamentals

1.3.3.2. Determining Factors

The parameters that determine the perception of similarity are numerous and their interaction might be very complex. However, a few important factors can be identified.

Inherent Features Due to Goldmeier [96], the degree of perceived visual similarity depends on the ratio of identical parts, as well as on the degree of variation of the non-identical parts. However, in addition to that he also mentions higher order features, such as the relationships of the parts, symmetries, and regularities.

Semantic Similarity Apart from the similarity between the actual stimuli, there is also a similarity between the meanings of stimuli (or the objects associated with the stimuli) which is in some cases in no way dependent on common physical elements [16]. Nevertheless these meanings—as additional features—may also influence the perceived similarity. For example an image depicting a conventional telephone may be rated similar to an image depicting a mobile phone although both images may not have much in common regarding the optical stimuli.

Selective Attention Composed stimuli might be perceived similar because they share some characteristics or features, whereas properties they do not have in common are neglected [16]. James [128] gave a simple example: The moon and a flame (gas-jet) are similar in respect of luminosity, whereas the moon and a football are similar in respect of rotundity.²⁰ For an example of this phenomenon with respect to perceived similarity of shapes, see Figure 1.8 on page 43.

It is argued, that not our complete knowledge of an object, but only a limited list of relevant features is used in a similarity assessment [222], and that these lists of features may even depend on the pair of objects under consideration [83]. Moreover, similarity judgements may even involve an active search for ways in which the two objects are similar, meaning that subjects search for features to justify high similarity ratings (see [210] cited in [138]).

It is also conceivable, that subjects do not only search for features, but also for ways of comparing them; that subjects choose the model which yields the highest similarity values. On the other hand, the active search might also

²⁰ As a flame and a football are not similar at all, this example is often invoked in order to question whether perceived dissimilarity fulfills the triangle inequality.

be performed in dissimilarity judgements. Two objects that have conspicuous features in common and differ in other conspicuous features, might—at the same time—get higher judgements of similarity and dissimilarity than two objects not having such conspicuous features (see also 1.3.3.3).

A special case of selective attention which deserves to be mentioned separately, is called *partial similarity*: the objects under consideration are divided into two complementary regions. The features contained in the first region are relevant for similarity assessment, and the features contained in the second region are almost ignored. An example which is often used to illustrate this phenomenon is a centaur (see, e. g., [230]), it is similar to a man because of its upper body and at the same time it is similar to a horse because of its lower body.

Context The assessment of similarity and dissimilarity of two objects is not only depending on these two objects alone, but also on extrinsic factors which are often subsumed under the term *context*. A general claim is, that any complete theory of similarity must account for context [15]. However, there are different types of such extrinsic factors affecting similarity judgements: stimuli noticed in connection with the perception of the actual objects under consideration, other objects under consideration, and the subject’s empirical knowledge.

The dependence on additional stimuli was, e. g., shown in [22]. In an experiment on judged similarity, two objects like, e. g., ‘flashlight’ and ‘rope’ got much higher ratings of similarity when they were presented with an additional caption such as ‘taken on camping trips’. This was explained with the context dependent activation (or increase of the weights) of properties such as ‘fits in a suitcase’²¹ that have influence on the similarity judgement.

The dependence on other objects under consideration was, e. g., shown in [223]. In an experiment subjects were to select from a list of three countries the one most similar to two reference countries. The following results (percentages of being chosen) were obtained:

Portugal + Spain:	France 45 %	Argentina 41 %	Brazil 14 %
Portugal + Spain:	France 18 %	Belgium 14 %	Brazil 68 %

According to the diagnosticity principle, the weight of a feature depends on its potential to be used for discriminating between objects. A feature that is shared by all the objects under consideration cannot be used to distinguish between these objects and has, therefore, relatively small influence on similarity judgements. When the set of objects under consideration is enlarged or

²¹ In the present example properties such as ‘needed to explore a cave’ or ‘useful in the wilderness’ are also conceivable.

1. Fundamentals

changed, such that the same feature is not shared by all the objects, then this feature acquires diagnostic value, gets a higher weight, and increases the similarity of the objects that share it [222].

The fact that France got significantly more votes than Brazil when presented together with Argentina, whereas France got significantly less votes than Brazil when presented together with Belgium, could be explained by such a different valuation of features. In the first triple, France is the only country also located in Europe, but language cannot be used to opt for either Argentina or Brazil. In the second triple, Brazil is the only country also using an Iberian Romance language, but the continent cannot be used to opt for either France or Belgium.

The dependence on the subject's empirical knowledge (*perspectival context*) was, e.g., shown in an experiment on judged similarity: Subjects that were either experts for clothing or experts for dogs rated the similarity between pairs of cloth and between pairs of dogs. To pairs from their own domain of expertise, the subjects gave lower similarity ratings ([210] cited in [97]).

All these examples refer rather to semantic similarity, but the inferences seem to be transferable to perceived similarity as well.

1.3.3.3. Properties

Regarding similarity, the aim of psychological research is to get a better understanding of how perceived similarity is judged and to find models that explain all observable phenomena. For automatically computed ratings of similarity on the other hand, deviations from reality are acceptable, as long as they are small. Therefore, in the following, the properties of perceived similarity reported by various researchers, are also rated according to their quantitative impact.

Range of Values, Conversion As perceived similarity cannot be quantified directly, the resulting range of values might be set almost arbitrarily. However, there are studies investigating the relation between perceived similarity and perceived dissimilarity. In [222], e.g., one group of subjects was asked to choose from two pairs of countries the pair being more *similar* than the other pair. Another group of subjects was asked to choose the pair more *dissimilar*. The sum of percentages of being chosen more similar and of being chosen more dissimilar was significantly greater than 100 for some pairs of countries, namely for the more prominent ones. This result is explained with different weightings of the common and the distinctive features in judgements of similarity and in judgements of dissimilarity.

Experiments with visual stimuli also confirm that perceived similarity and perceived dissimilarity are not complementary measures of the same psychological facts [162]: Stimuli were composed patterns either sharing more attributes, or sharing an internal relation between attributes. Subjects were asked to select the pattern more similar to, or the pattern more dissimilar from a reference pattern. Again the sum of percentages of being chosen more similar and of being chosen more dissimilar was significantly greater than 100 for some patterns, namely for the ones sharing an internal relation with the reference. Sharing internal relations seems to be a feature that is more important in similarity judgements, whereas the number of different attributes is more important in dissimilarity judgements.

However, in many cases the assumption that perceived similarity and perceived dissimilarity are complementary, meaning that there is a linear dependency with slope -1, is quite reasonable [222].

Metric Properties While mathematicians and computer scientists often prefer dissimilarity measures which fulfill the metric properties, among psychologists it is widely suspected that standard distance-based similarity measures do not provide an adequate account of perceived similarity [15]. An extensive examination of perceived dissimilarity with respect to the metric properties was presented in the heavily cited article ‘Features of Similarity’ by Tversky [222]. He actually comes to the conclusion that perceived dissimilarity is surely not a metric.

non-negativity As the range of values of perceived dissimilarity might be set almost arbitrarily, non-negativity by its own is virtually not confutable. However, if also small self-distance (meaning that the dissimilarity of an item to itself is 0) is assumed, there are indications that non-negativity does not hold: In recognition experiments often some off-diagonal recognition frequencies exceed the diagonal entries [222], which implies that some items are less dissimilar to a reference than the reference to itself.

small self-distance Also the claim that for every item the dissimilarity to itself is 0, conflicts with the results of recognition experiments: Often the frequencies of correct recognition differ from item to item [222].

isolation For isolation (meaning that if two items are not the same, their dissimilarity has to be greater than 0) the same arguments as for non-negativity hold: For its own it is virtually not confutable but in connection with small self-distance it is challenged by the results of recognition tasks. Moreover, according to the fact that sufficiently small differences are not noticeable (see 1.2.1), it is questionable whether isolation could be assumed at all.

1. Fundamentals

symmetry Linguistic usage knows basically two ways of making statements about similarity, namely a way differentiating roles: “object a is similar to object b ”, and a way not differentiating roles: “objects a and b are similar”. The claim of symmetry is, however, that the result does not change when both objects change the roles—in particular if roles *are* differentiated between.

Experiments on the similarity between countries showed that the judged similarity of country a to country b exceeded the judged similarity of country b to country a , if b was the more salient one [222]. In recognition tasks on shapes (polygons) performed by humans and by pigeons such asymmetries or response biases were also observed [168]. Moreover, investigating reference points (prototypes) in natural categories, it was observed that non prototypical stimuli were judged more similar to the prototype of their own category, than vice versa [194].

Obviously, perceived similarity is not inherently symmetric. However, the symmetry assumption seems to be adequate in many contexts [222].²²

triangle inequality Dissimilarity data derived from experiments can easily be transformed into a measure that fulfills the triangle inequality by adding a sufficiently large offset to any value for a pair of different objects [224]. However, it is widely suspected that perceived dissimilarity may sometimes violate the triangle inequality [15]. In connection with some standard assumptions, the triangle inequality implies properties that have been tested in [224]. The results of this study also suggest not to expect the triangle inequality to hold for perceived dissimilarity.

There are many examples that, although they do not formally disprove the triangle inequality, obviously seem to violate it. Most of these examples are based on selective attention, like the one given in [222]: Jamaica is similar to Cuba (because of geographical proximity); Cuba is similar to Russia (because of their political affinity)²³; but Jamaica and Russia are not similar at all. Figure 1.8 shows an example where the triangle inequality seems not to hold for the perceived similarity of shapes.

²² Please note, that trademark image retrieval, nevertheless, may require definitions of (dis-) similarity measures that are *not* symmetric (see 1.4.3.3).

²³ At the time the article was published, at least the Sowjet Union and Cuba used to be socialist states.

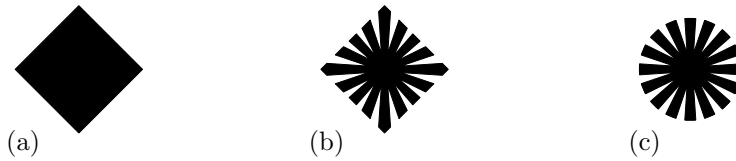


Figure 1.8. Shapes challenging the triangle inequality: Shapes (a) and (b) look similar because of their square-like global appearance and shapes (b) and (c) look similar because of their star-like inner structure, but shapes (a) and (c) seem to have nothing in common at all.

Since they are so counter intuitive, the violations of non-negativity, small self-distance, and isolation might be dismissed as being based on weak assumptions about the relation of perceived similarity and confusability. Moreover, the consequences in practice—also of non-symmetry—might be very limited. However, the violation of the triangle inequality is a substantive problem in practice.

Invariance In everyday life, objects have to be recognized even if the viewpoint changes. Therefore, the influence of the corresponding transformations on perceived similarity should be small. However, experiments are conceivable that guide the attention to the effects of the transformations and therefore, strict invariance of perceived similarity will be disprovable for virtually every class of transformations. In practice, on the other hand, it is important to know, whether the differences can be expected to be very small or whether they are substantial.

translation Due to the lack of a fixed reference point, in most cases, perceived similarity might be assumed to be invariant (with respect to synchronous transformation) to translations. In experiments where stimuli are presented side by side, even the relative positions will often not be considered as being features of the stimuli. In an experiment on judged similarity, where triangles were displayed simultaneously, a dependence on the relative positions, however, was observed [16]. It is liable that in this special case the variations were not caused by shifting the attention towards the relative position, but by mutual influence on the perception (of the slopes) of the triangles. Nevertheless, in most applications it is surely reasonable, to assume perceived similarity to be invariant (with respect to asynchronous transformation) to translations.

1. Fundamentals

uniform scaling Changes in the proximity of an object to the observer directly correspond to changes of the scaling. The size of the projection on the retina is first of all not perceived as an absolute feature of the object but only relative to other objects. If stimuli are presented consecutively, changes in size, therefore, have nearly no influence on perceived similarity [96].

rotation Although one might assume that perceived similarity is also invariant (with respect to synchronous transformation) to rotations, a counter example was given in [96]. Figure 1.9 shows a reconstruction of the stimuli. In the experiment the parallelogram (a1) was rated *less* similar to the square (a2) than the rectangle (a3) was, because the difference in slopes is more obvious than the difference in aspect ratios when the sides are almost horizontal and vertical. However, after rotation, the parallelogram (b1) was rated *more* similar to the square (b2) than the rectangle (b3) was.²⁴

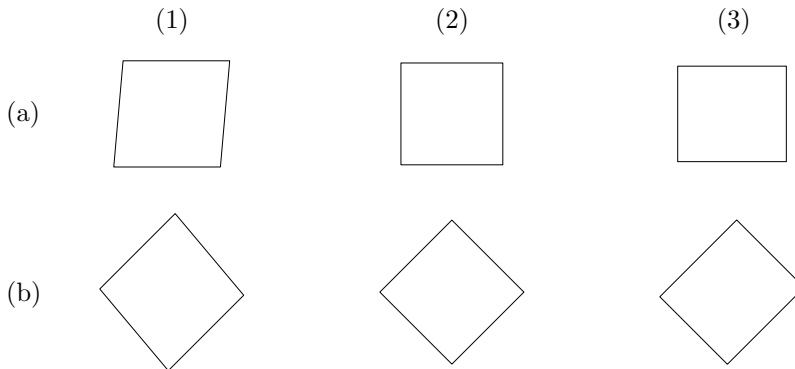


Figure 1.9. Shapes challenging rotation invariance:
Among the original shapes in row (a), the shape chosen as most similar to shape (a2) is usually shape (a3). Among the rotated shapes in row (b), the shape chosen as most similar to shape (b2) is usually shape (b1).

It seems to be obvious that perceived similarity is not invariant (with respect to asynchronous transformation) to rotations. Formally it can be deduced from the observations above or from the asymmetries due to different prototypicality of line orientations as reported in [194].

²⁴ It seems, that in the original experiment, the parallelogram (1) was rotated by 45° in clockwise direction, whereas the rectangle (3) was rotated by 45° in counterclockwise direction. However, the observations are expected to coincide.

General findings about the dependence of perceived similarity on the angle of rotation are hard to achieve, because the similarity also heavily depends on rotational and reflectional symmetries of the stimuli under consideration [96]. However, in most cases the influence of rotations on perceived similarity should be relatively small.

reflection For a discussion of strict invariance to reflections, the same arguments as for rotations hold—as the stimuli used there are axially symmetric, corresponding results can be achieved, by replacing the rotation with an adequate reflection.

In general, the influence of a reflection on perceived similarity heavily depends on the axis of reflection. The smallest changes are expected for reflections at vertical lines [158]. Concerning reflections at a vertical and at a horizontal line, a possible reason for the differences is given in [96]: a vertical axis separates regions that are conceptually equivalent (left and right which are even confused quite often) whereas a horizontal axis separates regions that are conceptually different (top and bottom). As a consequence, shapes mirrored at the vertical axis are very similar to the original—which was, e.g., also observed in [16].

Selectivity As discussed in Section 1.2.1, the ability to distinguish between simple stimuli that only differ slightly is limited and the magnitudes of the maximal differences that cannot be detected depend on the magnitude of the stimuli themselves. There is evidence, that also changes in the difference between two stimuli become less noticeable when the total difference between the stimuli is increased [16]. Analogously to Fechner’s claim about the perception of simple stimuli [79], a conceivable implication would be that perceived dissimilarity is also sublinearly depending on the difference of the stimuli. Figure 1.10 shows the results of an experiment which seem to confirm this assumption.

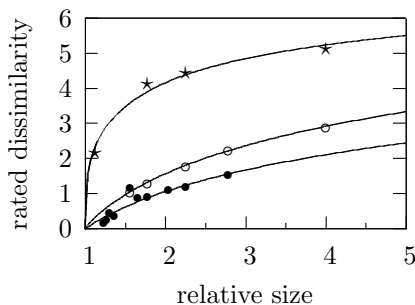


Figure 1.10. Size and dissimilarity: dissimilarity ratings for parallelograms (★), squares (◦), and triangles (●), plus fitting curves that are essentially logarithmic²⁵. Within each category the figures only varied in their sizes as measured by the area (raw data taken from [16]).

²⁵ parall.: $6.2 \cdot (\log_{10}(x))^{1/3}$, squares: $4.6 \cdot (\log_{10}(x))^{0.9}$, triangles: $3.5 \cdot \log_{10}(x)$

1.3.4. Some Examples of Measures and Models

There are mainly two types of (dis-)similarity measures. Firstly, measures motivated by mathematical considerations, which do not take the peculiarities of perceived similarity into account. Secondly, models motivated by psychological considerations, which are virtually not computable because of their complexity or because of their dependence on various unknown parameters. In [15] the attempts to bridge this gap were commented as follows: “*In many models, the process of adding simplifying assumptions is guided by mathematical simplicity rather than empirical validity*”.

As mentioned above, in the given context such simplifying assumptions are acceptable, as long as the resulting deviations from reality are small. However, some measures used in practice show a considerably different behavior. In this section a few selected measures of similarity and dissimilarity are presented and analyzed with respect to their mathematical properties and their applicability to the problem of emulating perceived (dis-)similarity. Further approaches to measure the (dis-)similarity of region shapes are described in Section 3.3.1.

Minkowski Distance Given two points $x, y \in \mathbb{R}^d$, their *Minkowski distance* of order $p \geq 1$ is defined as

$$\text{dist}_p(x, y) := \left(\sum_{i=1}^d |x[i] - y[i]|^p \right)^{1/p}.$$

Since the Minkowski distance can be defined via the L_p norm by $\text{dist}_p(x, y) = \|x - y\|_p$, it is often called L_p metric. The Minkowski distance of order 1 is also known as the *Manhattan distance*, and the Minkowski distance of order 2 is also known as the *Euclidean distance*. The limit of the Minkowski distance when p goes to infinity equals the so called *Tschebyscheff distance* or *maximum metric*:

$$\lim_{p \rightarrow \infty} \text{dist}_p(x, y) = \max_{1 \leq i \leq d} \left\{ |x[i] - y[i]| \right\}.$$

These distance measures either may be directly applied to estimate the dissimilarity between objects that are given as (usually high-dimensional) vectors of feature values, or may serve as the underlying distance²⁶ in the computation of measures of dissimilarity between sets of points (for example in the plane).

²⁶ If not stated differently, in the remainder of this work always the Euclidean distance will be used as underlying distance measure. In any case, it will be assumed, that the underlying distance measure can be computed in constant time.

As the name implies, any Minkowski distance of order ≥ 1 fulfills the metric properties (for $0 < p < 1$ the triangle inequality would be violated). Any Minkowski distance is invariant (with respect to synchronous transformation) to translations. Furthermore, the Euclidean distance is also invariant (with respect to synchronous transformation) to rotations and reflections.

It is also possible to define a *weighted Minkowski distance* as

$$\text{dist}_p^w(x, y) := \left(\sum_{i=1}^d w_i \cdot |x[i] - y[i]|^p \right)^{1/p} \quad \text{with } w_i \geq 0 \text{ and } \sum_{i=1}^d w_i = d.$$

This variant still fulfills the metric properties and is invariant (with respect to synchronous transformation) to translations, but the weighted Euclidean distance in general is not invariant to rotations and reflections anymore.

Dynamic Partial Function For objects that are given as vectors of feature values, a measure of dissimilarity which is based on the Minkowski distance, but gives better results with respect to perceived dissimilarity²⁷ was presented in [151]. The observation, that the way in which two objects are perceived similar is depending on the objects at hand (see also Section 1.3.3.2 on *selective attention*), led to the idea of dynamically selecting a limited number of dimensions for measuring the objects' dissimilarity with.

Given two vectors $x, y \in \mathbb{R}^d$, and a fixed positive integer $m \leq d$. Let $\Delta_i = |x[i] - y[i]|$ be the distance in dimension i and let $(\Delta_{\pi_1}, \dots, \Delta_{\pi_d})$ be the sequence of these distances ordered such that $\Delta_{\pi_i} \leq \Delta_{\pi_j}$ for $i < j$. Let furthermore $I = \{\pi_i | i \leq m\}$ be the indices of the m smallest distance values. Then the dynamic partial function of order p is defined as

$$\delta_{m,p}(x, y) := \left(\sum_{i \in I} |x[i] - y[i]|^p \right)^{1/p}.$$

As it violates the triangle inequality and the isolation criterion, the dynamic partial function is not a metric. However, it fulfills the non-negativity, small self-distance, and the symmetry criterion. The dynamic partial function is invariant (with respect to synchronous transformation) to translations, but—different from the Euclidean distance—not to rotations and reflections.

²⁷ In the original work actually not perceived dissimilarity but a predefined, unverified equivalence relation was used as reference for the experiments. See Section 1.6.1 for a discussion.

1. Fundamentals

Taking account of the characteristics of perceived (dis-)similarity that are caused by selective attention seems to be a step in the right direction. However, totally ignoring the $(d - m)$ worst dimensions may cause other problems: Whenever objects equal in m dimensions, their dissimilarity according to the dynamic partial function is 0. No matter how big the distance in the remaining dimensions are, the dynamic partial function is incapable of differentiating between the objects. Moreover, an object equaling a reference in m dimensions but having maximal distance in the remaining ones, will be rated more similar to the reference, than an object with only infinitesimal distance in all—or even $(d - m + 1)$ —dimensions.

Instead of multiplying the m best dimensions with 1 and the $(d - m)$ worst dimensions with 0 it is also conceivable to use a profile of dynamically assigned weights $(\omega_1, \dots, \omega_d)$ with $\omega_i > 0$ and $\sum_{i=1}^d \omega_i = d$. The resulting measure

$$\delta_{\omega,p}(x, y) := \left(\sum_{i=1}^d \omega_i \cdot |x[\pi_i] - y[\pi_i]|^p \right)^{1/p}$$

could still emulate selective attention, but on the other hand would fulfill the isolation criterion and avoid the counterintuitive behavior described above.

Hausdorff Distance Given two compact point sets $X, Y \subset \mathbb{R}^d$ and an underlying distance measure $dist: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, the *Hausdorff distance* between X and Y is defined as

$$d_H(X, Y) := \max(d_{\vec{H}}(X, Y), d_{\vec{H}}(Y, X))$$

with

$$d_{\vec{H}}(X, Y) := \max_{x \in X} \left\{ \min_{y \in Y} \{ dist(x, y) \} \right\}$$

being the *directed Hausdorff distance*. Informally, the Hausdorff distance measures to which extent each point of X lies near some point of Y and vice versa.

The *minimum Hausdorff distance* with respect to a class \mathcal{T} of transformations $t: \mathbb{R}^d \rightarrow \mathbb{R}^d$ is defined as

$$d_H^{\mathcal{T}}(X, Y) := \min_{t_1, t_2 \in \mathcal{T}} \left\{ d_H(t_1(X), t_2(Y)) \right\}.$$

The Hausdorff distance of sets of n and m points in \mathbb{R}^2 , as well as of sets of n and m non crossing straight line segments in \mathbb{R}^2 respectively, can be computed in time $O((n + m) \cdot \log(n + m))$ using Voronoi diagrams [6].

For sets of n and m points in \mathbb{R}^2 the minimum Hausdorff distance with respect to translations can be computed in time $O(nm \cdot \log^2(nm))$ if the underlying distance measure is either the Manhattan distance or the maximum metric [51], and in time $O(nm(n+m) \cdot \log(nm))$ for any other Minkowski distance [118]. For sets of n and m non crossing straight line segments in \mathbb{R}^2 the minimum Hausdorff distance with respect to translations can be computed in time $O((nm)^2 \cdot \alpha(nm))$ —with α being the inverse Ackermann function—if the underlying distance measure is either the Manhattan distance or the maximum metric [118], and in time $O((nm)^2 \cdot \log^3(nm))$ if the underlying distance measure is the Euclidean distance [2].

The Hausdorff distance fulfills the metric properties and it adopts invariances from the underlying distance measure. The Hausdorff distance is robust against deformation, blur, and crack. However, since its value is determined by a single point pair, the Hausdorff distance is not robust against noise.

As the Hausdorff distance does only consider the positions of points, it is a very generic measure and accepts any compact point set as input. On the other hand, it cannot take account of any internal structures or relationships between the points. These internal structures and relationships, however, may be essential for the perception of similarity. Apart from the lack of noise robustness, this is one reason why the Hausdorff distance may provide results, that conflict with human perception (see Figure 1.11 for an example). Nevertheless, the minimum Hausdorff distance has also been considered for the comparison of images (see, e. g., [119]).

Fréchet Distance A simple curve is a special set of points which is homeomorphic to a straight line segment. A distance measure that—unlike the Hausdorff distance—does take account of the continuous, one-dimensional nature of curves is the *Fréchet distance* (as introduced in [89]). Given two parameterized curves $f, g : [0, 1] \rightarrow \mathbb{R}^d$ and an underlying distance measure $dist : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, the Fréchet distance between f and g is defined as

$$d_F(f, g) := \inf_{\substack{\omega_1 : [0,1] \rightarrow [0,1] \\ \omega_2 : [0,1] \rightarrow [0,1]}} \left\{ \max_{t \in [0,1]} \{ dist(f(\omega_1(t)), g(\omega_2(t))) \} \right\}$$

where ω_1 and ω_2 range over continuous monotone increasing functions with $w_1(0) = w_2(0) = 0$ and $w_1(1) = w_2(1) = 1$. Informally, a common parameterization of f and g is looked for, such that the maximum distance at any time t is as small as possible [197]. Numerous variations of this problem have also been studied, e. g., a generalization to surfaces (see [95, 36]) and the so called *weak Fréchet distance* d_{wF} which is defined in the same way as the Fréchet distance, except that the reparametrizations ω_1 and ω_2 are *not* required to be monotone increasing.

1. Fundamentals

The *minimum Fréchet distance* with respect to a class \mathcal{T} of transformations $t : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is defined as

$$d_F^{\mathcal{T}}(f, g) := \min_{t_1, t_2 \in \mathcal{T}} \left\{ d_F(t_1 \circ f, t_2 \circ g) \right\}.$$

For two polygonal curves of n and m vertices the Fréchet distance can be computed in time $O(nm \cdot \log(nm))$ [5]. This result has also been generalized to piecewise smooth algebraic curves [197]. For two polygonal curves the minimum Fréchet distance with respect to translations can be computed in time $O((nm)^3(n+m)^2 \cdot \log(n+m))$ if the underlying distance measure is the Euclidean distance [7].

The Fréchet distance fulfills the metric properties and it adopts invariances from the underlying distance measure. The Fréchet distance is robust against deformation, but as it is defined on pairs of single curves, the terms blur robustness, crack robustness, and noise robustness are not applicable.

Since the Fréchet distance takes account of the courses of the curves, it is superior to the Hausdorff distance in discriminating between curves that are not perceived similar. On the other hand, patterns perceived similar may be formed by curves that consist of parts which are locally very similar (with respect to the Fréchet distance), but which are connected in different ways. In these cases, modelling the whole patterns by curves in combination with using the Fréchet distance is not compatible with human perception of similarity (see Figure 1.11 for an example).

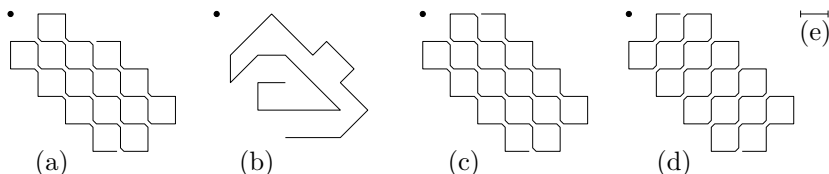


Figure 1.11. Applicability of Hausdorff distance and Fréchet distance: Assume that the line segment (e) has length 1, and that the origin is always marked by the dot. The Hausdorff distance between curves (a) and (c) equals 1 whereas the Hausdorff distance between curves (b) and (c) is only $\sqrt{1/2}$. The Fréchet distance between curves (a) and (c) on the other hand equals $\sqrt{2}$ whereas the Fréchet distance between curves (b) and (c) is much larger. However, the Fréchet distance between curves (c) and (d) is even greater than 2 and can be made arbitrarily large by such a construction.

Set-Theoretic Models Various models of similarity are based on the assumption that each object under consideration can adequately be represented by a set of discrete features. Given two such object representations A and B , approaches range from simply counting the number of common elements, to the so called *contrast model* $\sigma(A, B) := \alpha \cdot f(A \cap B) - \beta \cdot f(A \setminus B) - \gamma \cdot f(B \setminus A)$ with $\alpha, \beta, \gamma \geq 0$ as proposed by Tversky [222]. The latter can be brought into accordance with many observations that have been made in experiments on perceived similarity, however, the restriction to discrete features entails the fundamental problem of finding representations that are adequate for estimating perceived similarity.

Many stimuli are of gradual nature (e.g., color, length, curvature, etc.) and there is no obvious way to represent them by sets of discrete features. In addition, a discretization of perceivable stimuli usually leads to pairs of pairs of different features for which the influence on the perceived similarity is different.²⁸ Moreover, there might also be other influences. The perceived similarity of nonsense-syllables, e.g., does not only depend on the occurrence of common letters, but also on their position, and equality of relations between elements as, e.g., the height-width ratio of rectangles may be more important than the equality of the elements themselves [16].

Geometric Models Various models of similarity are based on the assumption that each object under consideration can adequately be represented by a fixed number of numerical values (which define a vector or point). The dissimilarity of two objects is then assumed to correspond to some value of distance between the two points. Such models can be used in two ways: either for analyzing existing dissimilarity data, or for deriving values of dissimilarity from the numerical representations (see also Section 3.3.1 page 138).

Given two such object representations $a, b \in \mathbb{R}^d$, according to the *weighted Euclidean multidimensional scaling model* the dissimilarity between a and b simply equals their weighted Minkowski distance of order 2. Various generalizations of this model have been proposed in order to achieve conformance with data from experiments on perceived similarity (see, e.g., [15] for an overview). In the *distance-density model* introduced in [138], e.g., the dissimilarity of the objects represented by the points a and b is assumed not only to depend on their distance d , but also on the spatial density of other object representations near a and b .

²⁸ In spoken English, e.g., a word containing the letter ‘v’ and the same word containing the letter ‘v’ instead, will be perceived more similar than words where ‘m’ has been replaced by ‘x’. In written English, on the other hand, a replacement of the letter ‘l’ by the numeral 1 sometimes is not even noticed (especially when typeset in a small font like used in footnotes) whereas replacing it by ‘w’ lowers the similarity significantly (‘replacement’ vs. ‘repwacement’).

1. Fundamentals

Transformational Models The basic assumption behind transformational models as stated in [121] is, that objects (patterns) are similar if they can be made identical by a few basic transformations. A well known measure of dissimilarity following this idea is the so called *Levenshtein distance* or *edit distance*: Given an alphabet Σ and two strings $s_1, s_2 \in \Sigma^*$, the Levenshtein distance $d_L(s_1, s_2)$ is defined as the minimum number of insertions, deletions, and replacements needed to transform s_1 into s_2 (cf. [150]). Instead of just counting the number of basic transformations it is also possible to define an arbitrary cost function on the set of transformations.

In addition to replacements, deletions, and insertions also other basic transformations are conceivable, e. g., rotations, reflections, and—if the patterns are of binary nature—inversion. Images might also be considered similar if colors are replaced by others, as long as relations of the colors stay the same [96].

Of course, the applicability of such a measure highly depends on the set of permitted basic transformations [121], and on the definition of the cost function. However, in connection with perceived similarity it is also very important to define under which conditions two patterns are considered equal. Demanding that the two patterns have to become identical is too restrictive as Figure 1.12 illustrates: For transforming a pattern into an arbitrary random pattern, no predefined set of basic transformations can yield a result essentially better than replacing every improper bit of information. However, the depicted patterns are perceived as very similar (if they are realized as being different at all)—a fact that was already pointed out in [17].



Figure 1.12. Random patterns:

(a) pseudo random pattern generated based on the first 2^{11} decimal digits of π , (b) random pattern generated based on 2^{11} random bits taken from [190].

The observation that perceived (dis-)similarity is not necessarily symmetric, namely, that non-prototypical stimuli are perceived more similar to prototypical stimuli than vice versa [194], can easily be brought into accordance with transformational models: *Normalizing* transformations (with respect to prototypicality) might be easier to perform cognitively, than their inverses [138]—discarding information about the variation from the basic form might be easier than adding such information.

Adaptive Resonance Theory and Beyond Mumford [168] sketched a scenario that seems to be a reasonable description of the processes involved in animal (and surely also human) object recognition: “A new scene or shape, after some pre-processing, first activates a set of features bottom-up. These features stimulate various higher-level categories of objects, and, in a top-down channel, templates of prototypes of these objects are produced. The lower-level tries to match these to the scene and this triggers new features describing the ‘residuals’, the mismatched features. Meanwhile the higher area also stores data on the range of allowable variations for each class of objects, and, on receiving these residuals, modifies the template reconstructions: the template is better thought of as a ‘flexible template’. [...] In this architecture, ‘similarity’ is totally customized to the type of object being recognized: for each category of object, the degree of similarity with a stimulus depends on how much variation is built into the template, and how strong are the features of the ultimate residual.”

Details of this scenario of course may be questioned, however, it gives an idea of how complex the processes probably are and how hard it probably is to correctly emulate the perception of similarity.

1.4. Retrieval

Generally speaking, *retrieval* is the process of searching a set of items for the ones having a property as specified by a given *query*. However, this process might be realized in very different ways, depending on the type of items a set may contain, and depending on the ways a query may be formulated.

1.4.1. Basic Concepts

Given a set S of n items and a query q , it is often assumed that the query uniquely partitions the set into two disjoint subsets: the so called *query set* S_q containing the items having the property queried for (relevant items), and the set $S_{\bar{q}} = S \setminus S_q$ containing the ones not having that property.

Based on S and q a *binary classification retrieval system* computes a *return set* S_r which should ideally equal the (normally) unknown query set S_q . Entities belonging to the return set S_r although not being relevant are called *false positives*. Entities not belonging to the return set S_r although being relevant are called *false negatives*.

1. Fundamentals

A retrieval system for images may make use of different kinds of information about the items in the set (see also [203]).

independent information Additional information that is not related to the contents of the images such as, e.g., their origin. It cannot be derived or inferred from the images only.

content-based information Information that can automatically be derived from the images. Content-based information typically describe images on a very low semantic level.

descriptive information Additional (typically externally provided) interpretations of the contents such as, e.g., which natural objects the image depicts. Till now, descriptive information mostly cannot be computed automatically, so usually it has to be gathered manually. The incapability to extract descriptive information automatically is often referred to as the *semantic gap*.

Retrieval using externally provided descriptive information is called *annotation based* retrieval (in contrast to *content-based* retrieval), even though the annotations may describe the contents of the images. Annotation based retrieval systems usually allow the formulation of Boolean queries of the kind “has the following annotation: ...”. Content-based retrieval systems on the other hand usually allow to search for images similar (with respect to some predefined measure of similarity) to an image provided by the user. This is called query by example (QBE) or similarity retrieval.

Normally, systems for similarity retrieval originally do not make a binary classification, but compute a ranking—a sorted sequence of the items—according to the similarity to the query item. A binary classification may then be derived from that ranking by taking a prefix of the ranked sequence as result set. The length of the prefix might either be fixed, or might depend on a threshold for the similarity.

Since automatic extraction of content-based information is often imprecise, content-based retrieval itself has to be robust against such imprecisions [203]. One way to make the retrieval more robust, is to use several measures of similarity based on different features which are extracted independently from each other, and to combine the results. This combination of results can be done either by normalizing the values of similarity in a predefined way and combining the resulting values (see [203] for an overview), by normalizing the values of similarity according to their actual distribution and combining the resulting values [13], or by computing the different rankings and then combining the ranks.

1.4.2. Retrieval Performance

The quality of a retrieval system is often estimated by applying it on sets and queries for which the query sets are known—so called *ground truth*. Typically these query sets have been compiled manually.

Given a set S and a query q , the effectiveness of a binary classification retrieval systems is mostly valued using the following measures or combinations of them:

Recall R The number of correctly returned items divided by the number of relevant items $R := \|S_r \cap S_q\| / \|S_q\|$.

Precision P The number of correctly returned items divided by the cardinality of the return set $P := \|S_r \cap S_q\| / \|S_r\|$.

Fallout F The number of false positives divided by the number of all non-relevant items $F := \|S_r \cap S_{\bar{q}}\| / \|S_{\bar{q}}\|$.

Given a ranking, the values for recall, precision, and fallout depend on the length k of the prefix that is taken as the result set. The recall $R(k)$ monotonically increases as the length of the prefix is increased, but the precision $P(k)$ typically tends to get smaller with increasing length of the prefix. The plot of the function that assigns to each length k the point $(R(k), P(k))$ is called the *precision-recall graph* and connecting consecutive points gives the *precision-recall curve*.

Let $n = \|S\|$ be the number of all items under consideration and $m = \|S_q\|$ be the number of all relevant items. Let $r(i)$ be the rank of the i^{th} -best ranked relevant item and $r_l := r(m)$ be the rank of the least-best ranked relevant item. The effectiveness of a retrieval system computing a ranking can be valued using the following measures as defined in [199, 72, 37].

Normalized Recall R_n Value in the range from 0 (worst case) to 1 (perfect retrieval). The normalized recall gives a higher weight to success in retrieving the first few items.

$$R_n := 1 - \frac{\sum_{i=1}^m r(i) - \sum_{i=1}^m i}{m(n - m)}$$

Normalized Precision P_n Value in the range from 0 (worst case) to 1 (perfect retrieval). The normalized precision gives equal weight to all retrievals.

$$P_n := 1 - \frac{\sum_{i=1}^m \log(r(i)) - \sum_{i=1}^m \log(i)}{\log\left(\frac{n!}{(n-m)! \cdot m!}\right)}$$

1. Fundamentals

Normalized Last Place L_n Value in the range from 0 (worst case) to 1 (perfect retrieval). The normalized last place indicates the cardinality of a result set (as obtained from a prefix of the ranking) which has reasonable expectation of containing all relevant items.

$$L_n := 1 - \frac{r_l - m}{n - m}$$

Average Precision P_a Value in the range from 0 (worst case) to 1 (perfect retrieval). The average precision is the mean of the precision values obtained after each relevant item is included in the result set when the length of the prefix is increased one by one.

$$P_a := \frac{1}{m} \sum_{i=1}^m \frac{i}{r(i)}$$

1.4.3. Trademark Image Retrieval

The application that motivated this research is the automated retrieval of trademark images: Given a set of trademark images and a query image (the *order* image), one wants to find all images within the set that are likely to get confused with the query image. The possibility of confusion occurs, whenever images as a whole look very similar, or when one image resembles a part of another image.

1.4.3.1. Services

Trademark images are usually registered as intellectual property of a company. Serving as cues for the company, its goods or services, the images have to be distinctive, non-confusable. Companies therefore have a big interest in making sure that their newly designed trademark images do not look similar to existing ones and, having trademark images in use, that no similar new trademark images are introduced by other companies.

Agencies having access to the trademark registries, offer respective services. Sifting through a whole database of trademark images so as to find the ones similar to a newly given order image is called a *trademark search*. Sifting through the newly incoming trademark images so as to find the ones similar to a given order image is called a *trademark watch* [202].

In many image retrieval scenarios it suffices to return at least some images fulfilling the conditions queried for. That is why the results provided, e.g., by *Google Images* [100] are often experienced as being very imposing. However, in trademark image retrieval one important objective is, *not to miss any* similar image.

1.4.3.2. Common Practice

As for databases containing hundreds of thousands of trademark images it is not feasible to manually compare an order to every image, the set of potentially similar images has to be restricted before manual inspection. One way to achieve this is to represent every image by a set of codes describing the image's content. The codes are typically assigned manually once a trademark is inserted into the database. Given an order, the database is then automatically searched for all images having one or more codes in common with the order image.

One of the most commonly used schemes for assigning codes to trademark images is the so called *Vienna Classification* [233] developed by the *World Intellectual Property Organization* [244]. The Vienna Classification defines a hierarchical system that divides figurative elements into 29 categories, 144 divisions, and 1 887 sections in total (state of fifth edition). A part of the hierarchy can be seen in Figure 1.13.

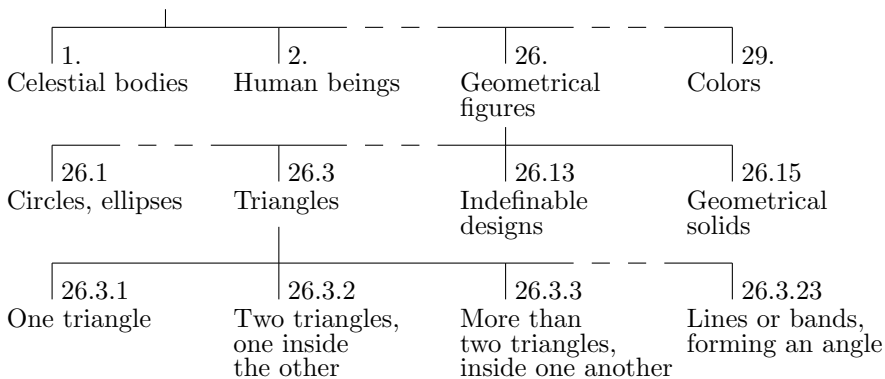


Figure 1.13. Part of the Vienna Classification

Using this classification scheme, the number of images that have to be inspected manually, can usually be reduced drastically for device marks depicting concrete objects such as, e. g., animals (category 3), household utensils (category 11), or musical instruments (category 22). However, many device marks are totally abstract and get classified as *other geometrical figures, indefinable designs* (code 26.13.25). Moreover some codes appear very frequently, such that for many searches several thousand images have to be considered.

1. Fundamentals

1.4.3.3. Automated Similarity Estimation

Due to the variety of ways in which a specific content might be depicted and due to the variety of ways in which two images might be perceived similar, with our present knowledge it is completely unrealistic to try and build a system that is able to handle *all* images correctly. Whatever the system looks like, there will always be images that have to be inspected by humans—which are the highest authority.

Instead of inaccurately handling all images—including also the difficult ones—and trying to slightly improve the rather poor quality, the goal should be to accurately handle most of the images and to reliably detect the difficult ones in order to sort them out. Improvements can then be made in increasing the number of images that actually can be handled.

With that fact in mind, one also may consider algorithms and techniques, that perform good on most images, but have an asymptotically bad worst case complexity—being able to automatically process a large fraction of the images in a database is actually a gain.

Possibilities to improve the trademark image retrieval by automatic support are

- reliably identifying very similar images,
- reliably identifying unsimilar images,
- speeding up the manual inspection by grouping the trademark images before the presentation to the trademark examiners,
- increasing the quality of the manual inspection (reducing the number of inconsistent decisions) by identifying duplicates within the database.

These different tasks, require different types of similarity measures: For an image to be rated very similar to a given query image, it suffices if it contains a part that almost looks like the query image, but it may also contain additional features (shapes, text, etc.). Therefore, a similarity measure σ^{cp} that considers the query image completely and the other image partially is needed. Duplicates on the other hand have to look almost the same completely. Therefore, a similarity measure σ^{cc} that considers the query image and the other image completely is needed.

1.4.3.4. Types of Images

Two types of trademark images can be differentiated: *Text marks* are images depicting only letters. The most important information contained in text marks are the words these letters form. Typeface, emphasis, or color are only of subordinate (or even of no) importance. *Device marks* on the other hand are images depicting no text, but graphic symbols, abstract graphics et cetera. For these kind of images shape is the most important feature. Color and texture are less important and are mostly only used to form the shapes.²⁹ Of course the boundary between text marks and device marks is not clear cut, textual and graphical elements might be mixed, single characters might be replaced by graphical elements, or letters might be distorted such that they become graphical elements rather than text (see Figure 1.14), but pure text marks need other treatment than pure device marks and for marks with mixed content a decomposition into textual elements and graphical elements would be desirable.



Figure 1.14. Text marks and device marks: artificial images depicting (a) the word ‘star’—classified as text mark, (b) the word ‘star’ with part of a letter replaced by a star (c) a star plus the abbreviation ‘sr’, (d) a distorted version of the abbreviation ‘sr’—classified as device mark

²⁹ Exceptions exist; sometimes even a color itself gets registered.

1.5. Significance of Heuristic Approaches

Generally speaking, a heuristic is a method that achieves *good* (but not necessarily *optimal*) results at low expense. However, which results are to be considered *good* heavily depends on the application at hand.

1.5.1. Basics

Let $f: I \rightarrow O$ be a problem specification that assigns to every input $i \in I$ a set $R(i) \subseteq O$ of possible results, and let $c: R(i) \rightarrow \mathbb{R}$ be a measure indicating the costs/quality of a result. For example if the problem is to find vertex covers³⁰ for graphs, I is the set of all graphs; for a given graph G , $R(G)$ is the set of vertex covers of G , and the cardinality of a vertex cover could be used as measure c .

If the task is to find for every input $i \in I$ the result $r_{\text{opt}}(i) \in R(i)$ such that $c(r_{\text{opt}}(i))$ is minimal (or maximal respectively), then f and c define an *optimization problem*. Given such an optimization problem it might be, that no algorithm computing an optimal result in reasonable time bounds is known. Many relevant problems are *NP-hard*, which means that they are strongly believed not to be computable in polynomial time at all—the example from above is such a case: Finding a vertex cover of minimum cardinality is NP-hard [56]. Moreover, for other optimization problems even the known polynomial time algorithms might not be applicable in practice because of too large exponents. In all these cases it might be helpful to have algorithms that do not necessarily find the optimal results, but results with specific properties.

Let $f: I \rightarrow O$ and $c: R(i) \rightarrow \mathbb{R}$ define a minimization problem and for a given input $i \in I$ let $r_{\text{opt}}(i)$ be the optimal result such that $c(r_{\text{opt}}(i)) \leq c(r)$ for any $r \in R(i)$. An algorithm A that for any given input i computes a result $r_A(i) \in R(i)$ such that $c(r_A(i)) \leq \alpha \cdot c(r_{\text{opt}}(i))$ for a fixed constant α , is called *constant factor approximation algorithm* with factor α (definition for maximization problems analogous). For the problem of finding a vertex cover for a given graph, e. g., there is a factor 2 approximation algorithm.³¹

Besides approximation algorithms there is another group of algorithms called *heuristics*. They do usually not give provable guarantees for every result they produce, but however, are used if they perform well on the average, or on most of the (reasonable) inputs. One example is the randomized quicksort algorithm as introduced in [110].³² In the worst case it has quadratic running

³⁰ A *vertex cover* for a graph $G = (V, E)$ is a set $C \subseteq V$ of vertices such that any edge $e \in E$ is incident to at least one vertex $v \in C$.

³¹ Greedily select an edge $e = \{u, v\}$, add both incident vertices u and v to the covering, remove all edges incident to u or v from the graph and recurse [56].

³² To make it fit the scheme above, one can consider it to output the sorted list plus the number of comparisons made, for using it as measure indicating the costs.

time, so there is no guarantee for every run, but the average running time is in $O(n \cdot \log(n))$ and the algorithm is widely used because of its good performance.

1.5.2. Heuristics in the Context of Similarity Estimation

It might seem surprising to mention, e.g., the Hausdorff distance in this context, as it is well studied and has many nice, provable properties. In fact it is really not a heuristic, however, not because of its provable properties, but because it is a problem definition, not an algorithm. On the other hand, using the Hausdorff distance to model perceived similarity is nothing more than a heuristic—one amongst others.

As long as it is not completely understood how humans rate similarity, the suitability of a given measure cannot be proved. On the contrary, for the Hausdorff distance—as for many other measures—it is easy to find examples, where it does absolutely not conform to perception. This does not mean, that it might not be useful for modelling perceived similarity in certain applications. However, the applicability of a given measure has to be confirmed based on its accordance to perceived similarity (rather than to other mathematical models) for expected real-world data.

1.5.3. No Free Lunch

There are many optimization problems for which no (applicable) algorithms are known that directly compute a result that is close to being optimal. To tackle such problems, a variety of search heuristics have been invented, such as, e.g., *hill climbing* in several variants (see [198]), *simulated annealing* [136], and *evolutionary algorithms* [192]. In order to be applicable to a whole range of different problems, these algorithms are often kept very generic, not making any assumptions on the problems at hand, which also leads to simple and transparent models.

In different tasks, the different generic search algorithms perform differently well. However, analyzing the general performance it was shown that averaged over all possible cost functions, all search algorithms (even a random search) exactly perform the same—a result which is called a *no free lunch theorem* [242, 243]. In other words: Whether a distinct search/optimization algorithm performs well, heavily depends on the cost function at hand.

There are many examples, where a generic search heuristic does actually yield good results, but according to the *no free lunch theorems* these are cases where the generic model (coincidentally) fits the optimization problem. As a consequence, it is suggested to explicitly tailor algorithms to particular problem classes by exploiting domain knowledge [187].

1.6. Data

As long as the ways in which perception and cognition work are not completely understood, the applicability of similarity measures to perceived similarity cannot be theoretically proved, but has to be confirmed in experiments using real-world data. Of course, the outcome of such experiments highly depends on the data used.

1.6.1. Requirements

There are mainly three reasons why experiments may produce unjustifiably good results:

- If the algorithm is tailored towards a specific data set instead of being tailored towards the general problem,
- if the used data set is tailored towards the algorithm,
- if the used data set enables algorithms to make right decisions for the wrong reason.

In order to eliminate these sources of misleading results, algorithms should be tested with real-world data, which cover the whole bandwidth and constitute a fairly representative cross-section of the data that occur in practice—a requirement that is often *not* fulfilled (see [184] for a detailed discussion).

Along with the data that serve as input for the algorithms, also information about the correct output is needed. This makes the generation of adequate data for testing algorithms with respect to their conformity with perceived similarity a laborious task: The values of similarity (or the information which items are perceived similar) have to be determined manually, which makes it almost impossible to get complete ratings for large data sets.

Automatically generated data may be used additionally, e. g., for the purpose of getting a first, rough idea about the performance of an algorithm, or to systematically search for limitations. However, care has to be taken to avoid misinterpretations of the results. In [46], e. g., from each image of a set of 60 000 images, 24 variants were generated by applying transformations such as rotations, scalings, cropping, and downsampling. The resulting sets of 25 images each, were supposed to be perceived similar—a strategy by which “*individual’s subjectivity*” was intended to be “*safely excluded*”.³³ A strategy which is very questionable, because it is targeted on identifying (transformed) duplicates of an image but has no verifiable connection to perceived similarity at all.

³³ The same data apparently also served as test set for the evaluation of the *dynamic partial function* in [151]

1.6.2. Employed Data Sets

Throughout this work mainly the following sets of data have been used to test the performance of the algorithms:

1.6.2.1. MPEG 7 CE-Shape-1 part B

The Motion Picture Expert Group (MPEG) [167], a working group of ISO/IEC, provided a data set that consists of 1 400 (mostly) silhouette bi-level black-and-white images—the MPEG-7 core experiment CE-Shape-1 part B set. The set is subdivided into 70 classes containing 20 related images each. The images of some classes depict silhouettes of real objects, such as apples, birds, or cars. Other classes contain totally abstract images. Figure 1.15 shows some examples.

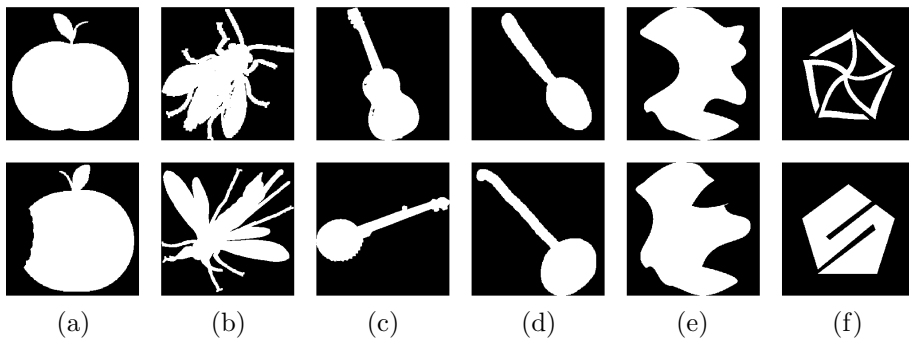


Figure 1.15. MPEG 7 images: examples of shapes from 6 different classes, namely (a) ‘apple’, (b) ‘fly’, (c) ‘guitar’, (d) ‘spoon’, (e) ‘mask’, (f) ‘device6’

For every image, the other 19 related images of the same class are supposed to be the ones most similar to that image meaning that all 20 images of the class are relevant in a similarity retrieval. Therefore, it is possible to perform 1 400 queries, one for each image. After the compilation of the set, the perceived similarities between the images have not been determined—at least not to the knowledge of the author. The set is therefore actually dedicated to be used in classification tasks, rather than in similarity estimation tasks. In addition, in some classes also semantics play a role. Nevertheless, the set serves as a good starting point for the evaluation of similarity measures. Moreover, it is commonly used to assess the performance of shape descriptors and retrieval systems and therefore there is quite a large number of published results for this set (see, e. g., [32] and [143]).

1. Fundamentals

1.6.2.2. Trademark Images from the UK Trade Marks Registry

The UK Trade Marks Registry provided a data set that consists of 10 745 bi-level black-and-white trademark image files—mostly device only marks. Figure 1.16 shows some examples.

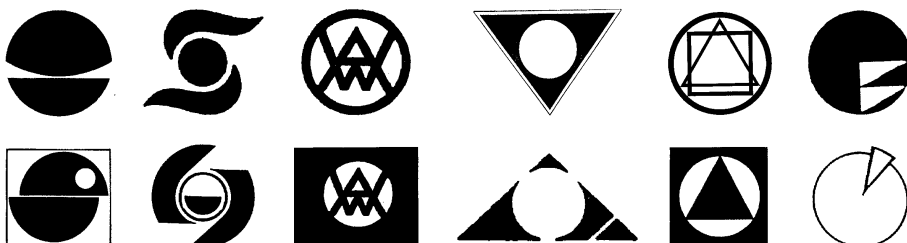


Figure 1.16. UK trademarks:
The first row contains 6 of the query images, the second row contains corresponding relevant images.

Most of the images depict abstract geometrical shapes. Some trademark logos are depicted in several image files and in several variations. Most of the images depict black figures on white background, but in some images the figures are hatched or have texture. For about 2 000 image files (19%) the number of closed contours (distinguishable black and white areas) exceeds 100. For about 800 image files (7%) the number exceeds 1 000 and the maximum observed is even 92 436. The images are supposed to depict a single trademark each. Some of the images however contain several versions of the same trademark. For a very low percentage of the images it is even hard to say what they contain at all (Some extreme examples are shown in Figure A.1 on page 173).

A set of 24 image queries is used as ground truth. Each query consists of a query image (out of the collection of 10 745 images) and a list of relevant image files³⁴ from the test set (including the query image file itself). The lists of relevant image files had been compiled by experienced trademark examiners.³⁵

³⁴ Relevant means that trademark examiners judged an image sufficiently similar to the query image to warrant detailed examination, not that infringement had necessarily taken place [72].

³⁵ First, the trademark examiners compiled initial lists. Then, with an early version of the *ARTISAN* retrieval system the test set was searched for images similar to the query images. The trademark examiners judged whether the automatically retrieved images were indeed sufficiently similar to the respective query image. If so, the list of relevant image files was augmented [72].

The 24 queries contain 333 image files in total. A complete listing of the query images and the respective relevant images is given in Table A.2 on page 176. The set together with the ground truth was also used to test the *ARTISAN* retrieval system [72].

Some decisions of the trademark examiners whether an image should be contained in the list of relevant images or not, are hard to comprehend for an outsider (Some examples are shown in Figures A.2 and A.3 on page 174). But as the examiners have the expertise, their judgement has to be taken as reference. However, the lists of relevant images are obviously not error-free:

- For 4 queries there exist image files depicting the original query trademark (copy of original), that are not contained in the lists of relevant image files: in total 5 additional image files.
- For 9 queries there exist image files depicting one of the relevant trademarks (copy of relevant), that are not contained in the lists of relevant image files: in total 30 additional image files contributed by 14 different trademarks.

Some of these inconsistencies might be caused by the later insertion of relevant image files to the list. However, the fact that even image files depicting the original query trademark were overlooked by experts (supported by a Vienna Classification based system) emphasizes the relevance of a reliable automated solution.

The experiments described in the present work were assessed based on the original queries as well as on the queries where the lists of relevant images have been corrected. Since the differences of the results were marginal, only the results based on the original queries will be listed.

1.6.2.3. Trademark Images from Aktor Knowledge Technology

The company Aktor Knowledge Technology provided a data set that consists of 20 894 trademark image files in total—device only marks as well as marks containing graphical plus textual elements. Unlike the *MPEG 7* data set and the *UK trademarks* set, this set also contains color images: Based on the segmentation described in Section 2.2.2, 38 % of the images (7 908) have been classified as pure bi-level black-and-white images, another 2 % (486 images) have also been classified as bi-level black-and-white images although containing different gray levels due to compression artefacts and blurring, 19 % (3 974 images) have been classified as gray level images, and 41 % (8 526 images) as color images. Details of some images from this set can be seen in Figures 1.5 (b), 2.4, and 2.5.

Extraction of Shapes

This chapter deals with the problem of extracting the perceptually relevant shapes from figurative images given as raster graphics. Two images may be perceived similar, just because of the fact that they contain a specific shape. For the retrieval of photographic images, shapes may be represented implicitly, using features or interest points based on local analysis of color distributions. Approaches going in this direction are, e. g., corner detection (see [60] for an overview) or the *scale-invariant feature transform (SIFT)* (see [155] and [156]). In figurative images, however, the shapes may be depicted in different ways and therefore it is essential to explicitly extract them.

In the following, the main aspects that have to be considered in automated image processing are outlined one by one, and approaches for solving the problems are developed. All the proposed algorithms have been implemented and in Section 4.2 they are joint together in a framework for automated extraction of shapes in figurative images.

2.1. Basics

Since figurative images are artificially designed, the extraction of shapes is—unlike in photographic images—normally not thwarted by, e.g., gradients of brightness or vague edges. However, shapes cannot be extracted in a uniform way, because of the different possibilities to depict them.

region vs. outline When a shape shall be depicted by its outline, any image representation that is suitable for visual perception needs to depict this outline by one or more regions of some width. Since the 'line-width' may vary from image to image and even from image representation to image representation, and since line width is also subject to the design of figurative images, there is no way to reliably discriminate between regions depicting a region shape, and regions depicting the outline of a shape (see Figure 2.1 (a) for an example).

textured regions When a shape is depicted by a textured region, the shape extracted from the image should—of course—equal the original shape and abstract from the details of the texture. Since there are no limitations on the granularity of the texture, there is no way to reliably discriminate between small regions¹ forming the texture of the shape actually depicted, and small regions that really depict small shapes (see Figure 2.1 (b) for an example).

hatched regions When a shape is depicted by its outline plus a hatched interior, the outline should to be discriminated from the lines that are just hatching. One forms the shape, the others might totally be neglected (see Figure 2.1 (c) for an example).

textured lines When a shape is depicted by its outline, even this outline might be textured, e.g., using dots, using small line segments, or a mixture of both (see Figure 2.1 (d) for an example).

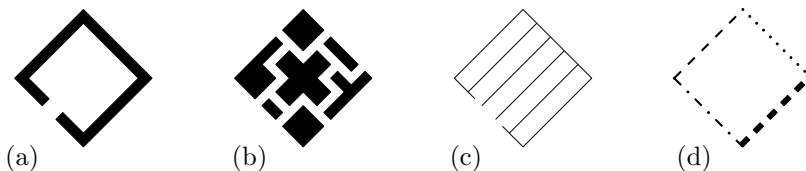


Figure 2.1. Challenges due to design:
(a) square or sharply bent strip, (b) textured square or cross etc., (c) hatched square or comb, (d) square depicted by varying discontinuous outline.

In addition to these challenges due to the design of figurative images, the representation as raster graphics and the use of lossy compression involve further difficulties in correctly extracting the depicted shapes.

dithering In order to create the illusion of gray tones in bi-level black-and-white raster graphics, black and white pixels may be blended—a technique called dithering (see Figure 1.1 on page 20 for an example). Moreover, patterns that consist of many different colors but are perceived as homogeneous regions are sometimes even used in color images (see Figure 2.5 for an example). In the following, this variant will be called *color disturbance*.

antialiasing Where lines or boundaries of regions do not run parallel to the horizontal or vertical axis, rasterization causes these lines and boundaries to appear jagged. Antialiasing is the attempt to reduce the impact of rasterization on perception. Typically, this is done by interpolating the pixels' colors depending on their nearness to the line (see Figure 2.2 for an example) or on their relative affiliations to the different regions, respectively. Moreover, blurring of the boundaries between regions of different colors is sometimes also used as a stylistic device.

compression artefacts To save storage or bandwidth, raster graphics typically get compressed. Although loss-less techniques are available, a lossy variant of the jpeg-compression [125] is often used even for figurative images. This variant is quite adequate in some domains, but for images with regions of homogeneous color and sharp boundaries between these regions it is rather unsuitable (see Figure 2.2 for an example).

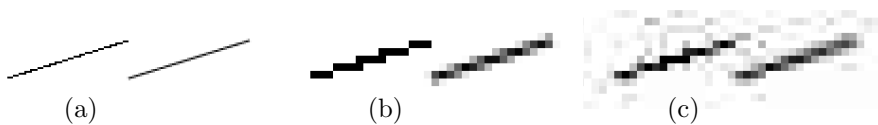


Figure 2.2. Challenges due to representation:
 (a) line segment rastered without and with antialiasing, (b) same line segments rastered at different resolution, (c) same line segments rastered at different resolution and jpeg-compressed with low quality.

¹ In general the regions forming texture need not be small, of course. Any shape or set of shapes filling the interior of the actually depicted shape sufficiently dense with its boundary can act as texture.

2.2. Vectorization of Figurative Images

For figurative images vectorization is essentially the reverse of the rasterization: deducing shapes from the pixels' colors.

2.2.1. Related Work

There are two opposing strategies for extracting shapes in images. The first one is to determine the shapes' interiors by looking for pixels that—according to some criteria of uniformity—are presumably belonging to the same shape, and to group these pixels together (see [104] for a comprehensive overview). Approaches following this idea are, e. g., thresholding, region growing (see, e. g., [254]), or mean shift analysis [55]. The other strategy is to determine the shapes' boundaries by looking for discontinuities. Two of the most widely used approaches of this type are the Canny edge detection algorithm as presented in [39], and the watershed transform [27] (see also [226]). Since uniformity and discontinuity are inversely related, also combined methods have been proposed such as energy minimization based on the Mumford-Shah functional as defined in [170], and some graph based approaches (see, e. g., [80]).

2.2.2. Discretization of Colors

Based on the observation that most figurative images make use of only a few distinct colors², the strategy pursued here is to identify these colors and to assign each pixel to one of these colors first (segmentation), and to detect the boundaries between regions of different color afterwards. The proposed approach, therefore, belongs to the first category of approaches mentioned above. Smooth color gradients like, e. g., in Figure 1.1 (page 20) cannot be handled properly this way, but the advantages of the selected strategy surpass this deficiency, which only occurs occasionally.

Idea of the Proposed Approach Many figurative images are given as (almost) bi-level black-and-white images, meaning that the colors occurring can be grouped into two clearly distinguished groups and that the variation within each group is not perceptible. For these images the discretization is trivial and they may easily be identified before more involved analysis is carried out, just by looking at the color histograms. Regions dithered with black and white

² However, due to compression artefacts, antialiasing, and color disturbance, the number of colors in a specific representation of such an image may be arbitrarily large.

pixels cannot be handled properly (meaning that they are classified as kind of homogeneous region of gray color) this way, but see Section 2.3.1 on texture analysis for this issue.

The basic strategy for the remaining images is to successively identify pixels for which a classification based on its color and context can be done with minimal risk of making wrong decisions. The whole process is subdivided into several steps. At first, the obvious cases are handled with moderate computational effort and the fewer unclassified pixels remain, the more detailed the analysis gets.

The proposed approach takes the following considerations into account:

- In regions with a high degree of variations (regions of color disturbance), pixels are not perceived as being individual, whereas in homogeneous regions even isolated pixels of different color may lead to the perception of some structure.
- The usage of blurring as a stylistic device, antialiasing, and compression artefacts may cause pixels at the boundary between two regions of different color to have an intermediate color. These pixels are rather perceived as belonging to one of the two regions, whereas in other areas of the image pixels with the same intermediate color may be perceived as individual structure.
- Near the boundary between two regions with colors that are perceived very differently, comparatively small differences are rather ignored (even when the additional colors are not intermediate ones), whereas in homogeneous regions even small differences may lead to the perception of some structure.

Description of the Algorithm The segmentation does not follow a single paradigm, but combines many ideas that appear to be helpful to overcome the obstructions one is confronted with in many real-world trademark images. An elaborate analysis of the performance and a systematic evaluation of optimal parameters and combinations of techniques is desirable, however, it is not the main focus of this work. In the following an overview of the ideas used to obtain a segmentation that can form a proper basis for the subsequent stages is given. A more detailed description of the algorithm is given in the Appendix (Section B.1).

2. *Extraction of Shapes*

The discretization of colors is done in 12 steps:

- 1st step: region growing** The goal of the first step is to identify large regions that have virtually no variation in perceived color, and to beware of all unsafe areas such as, e. g., boundaries. One after the other, clusters are grown greedily, but in a very cautious way, avoiding all pixels that are not far away from any sensible change in color.
- 2nd step: enlarging and merging clusters** Since in the first step the clusters were grown one after the other this had to be done very cautiously. In the second step the existing clusters are further enlarged concurrently and—if possible—merged. The whole process of enlarging the clusters and merging the clusters is repeated until no further changes occur.
- 3rd step: detecting border pixels** In the first two steps, only pixels in homogeneous regions have been processed, pixels near significant changes of color have been excluded. Two types of these pixels can be differentiated between: first, pixels actually having almost the color of the shape they depict; second, pixels located between two different regions and having a color that is a mixture of these two regions' colors, e. g., occurring in blurred images. Assigning the latter pixels to such intermediate colors would result in the detection of shapes that have no perceptual counterpart—they should rather be assigned to one of the colors of the regions they are located between. Therefore, for every unclassified pixel the degree of being a border pixel is computed.
- 4th step: further enlarging and merging clusters** The same enlarging of clusters and merging of clusters as in step 2 is performed, however, whether a pixel might be added to a cluster, now does also depend on the pixel's degree of being a border pixel.
- 5th step: uniqueness** Unclassified pixels that have a color which differs a lot from the colors of all the clusters in the vicinity more likely originate from an independent colored region than pixels with colors only slightly differing from the clusters in the vicinity do. In order to distinguish between both cases, for every unclassified pixel its uniqueness—the degree of having a color independent from existing clusters in the vicinity—is computed.

- 6th step: merging** In steps 1 to 4, only clusters that were grown from homogeneous regions (although also reaching for pixels near abrupt color changes) have been considered. Now, in order also to find thin clusters as resulting from the depiction of lines, every unclassified pixel is seed of a (preliminary) cluster and neighboring clusters of sufficiently similar color are merged. However, since border pixels should not result in independent clusters, and since pixels in the vicinity of other clusters that they might belong to should not result in independent clusters, the merging also depends on the borderiness and on the uniqueness of the pixels. After the merging, every pixel is member of a (preliminary) cluster, however, not every preliminary cluster corresponds to an independent perceptual entity—only sufficiently large clusters can be assumed to be valid. Therefore, all pixels belonging to small clusters are marked as unclassified again.
- 7th step: further enlarging and merging clusters** After additional clusters possibly have been introduced, the clusters again are enlarged and merged as in step 4.
- 8th step: merging clusters based on color** In order to reduce the number of distinct colors, also clusters that do not share common edges are merged.
- 9th step: handling antialiasing** Pixels that are located inbetween two different regions (clusters) and that have an intermediate color ideally should be assigned to one of the two regions. In order to do so, for every unclassified pixel its vicinity—similarly to step 3—is searched for evidences that the pixel is an intermediate one. If enough evidence is found the pixel is added to the cluster for which essentially the difference in color is smaller.
- 10th step: aggressive assignment** Similarly to step 4, the clusters are again enlarged, however, since border pixels are already processed, the tolerated differences in color are larger.
- 11th step: merging clusters based on color again** Again, clusters are merged based on their colors.
- 12th step: clustering rest** The remaining pixels are clustered just based on their distribution in color space. Repeatedly the largest cluster is determined until no unclassified pixel remains.

2. Extraction of Shapes

Experimental Results Using an implementation with the set of parameters described in the Appendix (Section B.1), the following results have been achieved: Figure 2.3 shows the result of the segmentation for an exemplary single image demonstrating several features of the segmentation process: A region of homogeneous color with clear cut edges (a) is recognized as such and left unchanged. Regions of homogeneous color with antialiased edges (c) are recognized as such and the pixels of intermediate color are assigned one of the colors of the adjacent regions. Pixels of the same intermediate color, but not located inbetween different colors are recognized as filigree structure (b) or as region (c). A region of color disturbance (e) is recognized as such and assigned a single color. Moderate jpeg-artefacts are eliminated, whereas isolated noisy pixels are left unchanged.

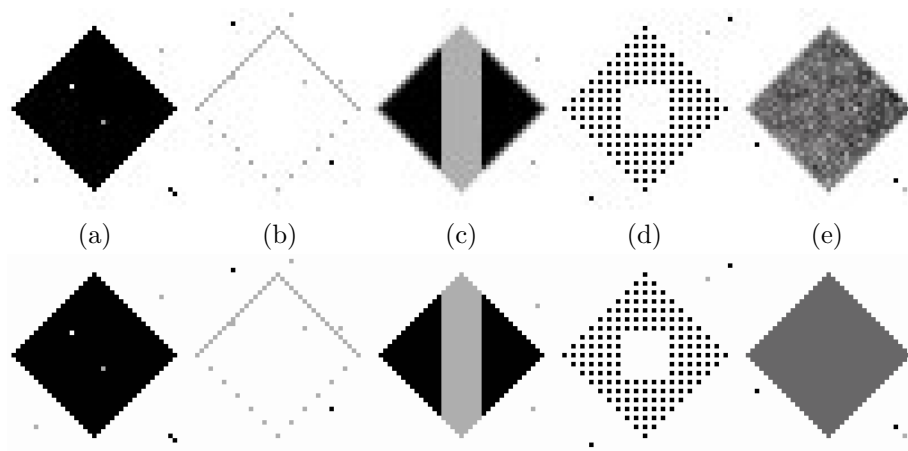


Figure 2.3. Segmentation—artificial image: original image (top) and its segmentation (bottom). All at the same time, the image contains region shapes, lines, dotted lines, antialiased edges, texture, and color disturbance, as well as noisy pixels and jpeg-artefacts³.

³ In a printed version of this thesis, the jpeg-artefacts might be hard to perceive due to dithering. However, viewed on a screen they are obvious.

2.2. Vectorization of Figurative Images

Figures 2.4 and 2.5 show the results of the segmentation for two real-world trademark images from the *Aktor set* (see Section 1.6.2.3). Figure 2.4 demonstrates the importance of a special handling of pixels at the boundaries: The difference between the colors of pixels belonging to the same region may be larger than the differences between the colors of pixels from different regions. In the lower part, the colors on the edges are almost interpolated, whereas in the upper part, there are also boundary pixels which do not have an intermediate color. Figure 2.5 shows the massive differences in color that may occur within a region that is actually perceived as having a single color.



Figure 2.4. Segmentation—trademark image I: original (left) and segmented (right) 90×60 pixel detail of a 510×1856 pixel trademark image. The average RGB values of the red region in the upper part are $(250, 8, 3)$ whereas the boundary pixels marked by the black circles have RGB values $(255, 255, 222)$, $(192, 43, 45)$, and $(227, 255, 252)$, respectively.



Figure 2.5. Segmentation—trademark image II: original (left) and segmented (right) 90×60 pixel detail of a 967×301 pixel trademark image.

2.2.3. **Boundary Detection**

After the image has been segmented, meaning that each pixel has been assigned to a color of the limited palette, ideally every depicted shape corresponds to a maximal edge-connected set of pixels of homogeneous color. For the process of extracting boundary polylines from the grid of pixels, every pixel (i, j) is seen as covering the square $[i, i + 1[\times [j, j + 1[$.

Having regions of homogeneous color, the determination of boundaries is then almost trivial: every line segment separating two pixels of different color is part of a boundary and for an edge-connected region of homogeneous color its boundary segments form a closed polyline. These closed polylines can be detected by

1. scanning the image for a pair of neighboring pixels having different colors,
2. following the discovered boundary until the starting point is reached again, while gathering all passed vertices, and
3. marking the boundary as being processed, and continuing with the scan.

The image is scanned row by row. Whenever a pixel having a neighbor with different color is found and the (half) edge separating the pixel from its neighbor is not already marked as being processed, the boundary that separates the pixel's edge-connected region of homogeneous color from pixels of other colors is followed until the starting boundary edge is reached again, marking every traversed half edge. If the detected boundary is an outer boundary (decided based on the sum of angles between boundary edges which is either $+2\pi$ or -2π) it is added to the list of shape boundaries. If the detected boundary is an inner boundary (bounding a hole from the outside) it is rejected since it will completely be represented by the outer boundaries of the hole's content.

In the scan every pixel is only considered once. Every edge is traversed at most twice (once for every half edge). Since the number of edges is linear in the number of pixels, for a raster graphic of size $w \times h$ the whole process of detecting boundary polygons between regions of different color can be completed in time $O(w \cdot h)$.

2.3. Merging of Small Shapes

The set of shapes resulting from the vectorization of a figurative image may contain shapes that—in the image—are perceived as texture forming the interior of another region shape, or as small patches forming a line shape (see Figure 2.3 (d) and (b) for examples), rather than being perceived as individual entities. Ideally, in both cases these shapes should be replaced by what they form, i. e., the actually perceived region shape in the first case, and the actually perceived line shape in the second case.

2.3.1. Textured Regions

In general, the term *texture* is used for the appearance of the surface of objects, including fine grained differences in brightness due to coarseness of the surface, as well as differences in color. Apart from color disturbance, in figurative images the first aspect is mostly only of limited relevance. Moreover, fine grained slight variations are assumed to be eliminated to a large extent already in the vectorization phase. This section therefore concentrates on texture with obvious variations in color due to dithering or stylistic methods used by the designers of images.

Related Work An extensive overview of the literature dealing with texture analysis is given in [221]: Approaches proposed in this context include statistical methods like using *gray level co-occurrence matrices* (see, e. g., [255]), geometrical methods like looking for placement patterns of some structural elements (see, e. g., [253]), and using *Voronoi diagrams* (see, e. g., [220]), signal processing methods like *Fourier analysis* (see, e. g., [248]), and *Gabor filters* (see, e. g., [126]).

For the problem of replacing texture in figurative images by the shapes that are formed, the approach used in [73] and in [113] was essentially blurring the image and applying edge detection to the blurred image. However, a blurring that is sufficient for equalizing the differences in textured regions, at the same time may destroy thin structures. This method should, therefore, not be applied to images that depict some shapes by textured regions and some shapes by their outlines.

2. Extraction of Shapes

Idea of the Proposed Approach In figurative images, detailed drawings of real-world objects may be as complex as texture. Due to the influence of semantics, automated systems may often not estimate the perceived similarity of such drawings properly. Therefore, the goal pursued here is, in the first place, only to detect regions of high visual complexity reliably. After that, the shapes inside this regions may either be replaced by the shapes they actually form, or the image may be marked for manual inspection (see Section 1.4.3.3).

The proposed approach differs from most of the existing methods due to the fact that it is not based on an analysis of the pixel colors (intensities), but on the boundaries extracted in the vectorization. Unlike in geometrical methods, however, the boundaries are not treated as individual entities, but are analyzed using techniques related to signal processing methods. The basic idea is to use a low-pass filter on the image of edges to identify regions of high visual complexity: The boundaries detected in the vectorization phase are drawn into an initially empty image. Applying a low pass filter to this image transforms regions containing lots of edges into regions of high intensity, and regions containing only a few isolated edges into regions of low intensity. The existence of regions that—in the original image—are textured or contain detailed drawings, may then be detected by applying thresholding to the intensity image. Moreover, using appropriate values for the blurring and for the thresholding, such an intensity image may also be used to estimate the boundaries of a textured shape, however it may not be used to distinguish between regions of different textures having the same density of edges.

Description of the Proposed Algorithm A gray level image of the same size as the original image is initialized with all pixels set to 0 (black). The boundary edges detected in the vectorization phase are then drawn into this image with maximal brightness (white).⁴ The low-pass filter applied here is a convolution with a two-dimensional rectangular function with side lengths d_b . Thresholding the smoothed image, again may result in a scattered set R_+ of pixels with brightness above a threshold θ_t . In order to counter this discretization effect, small holes should be closed and small isolated groups of pixels, which might also result from the crossing of jagged lines in the original image, should be removed. This is achieved by applying two morphological operators on R_+ , namely a *closing* and an *opening*.

⁴ Actually, the pixels that have the lower left corner covered by an edge are set to the maximal brightness. As the edges pass between the pixels of the original image, this re-drawing causes a shift by 0.5 in positive x-direction and in positive y-direction. However, for detecting regions of high complexity this shift, of course, is insignificant.

Let A be a subset of the \mathbb{R}^2 . The *dilation* of A by a disc D of radius r , centered at the origin, is—analogically to the Minkowski sum—defined as $A \oplus D := \{a + v \mid a \in A, v \in D\}$. It corresponds to pushing the boundaries towards the outside by r . The *erosion* of A by D is defined as $A \ominus D := \{p \mid \nexists a' \in \mathbb{R}^2 \setminus A, v \in D \text{ such that } p = a' + v\} = (A^c \oplus D)^c$.⁵ It corresponds to pushing the boundaries towards the inside by r . The *closing* of A by a disc D is a dilation followed by an erosion, namely $(A \oplus D) \ominus D$. The *opening* of A by a disc D is an erosion followed by a dilation, namely $(A \ominus D) \oplus D$.

In case of raster graphics, the underlying space is not \mathbb{R}^2 but \mathbb{Z}^2 so formally, the disc D has to be replaced by $D' = D \cap \mathbb{Z}^2$. The union R_t of regions that are assumed to contain texture or detailed drawings is derived from R_+ by applying a closing and an opening: $R_t = (((R_+ \oplus D') \ominus D') \ominus D') \oplus D'$. If $R_t = \emptyset$, there are no relevant regions of high intensity which indicates, that there are no regions of high visual complexity. If R_t is not the empty set, meaning that regions of high visual complexity do exist, either the image may be marked for visual inspection, or further texture analysis techniques as listed above may be applied to the parts of the image covered by R_t .

Experimental Results Using an implementation with the set of parameters described in the Appendix (Section B.1), the following results have been achieved: Figure 2.6 shows the result of the texture detection for the exemplary image from Figure 2.3.

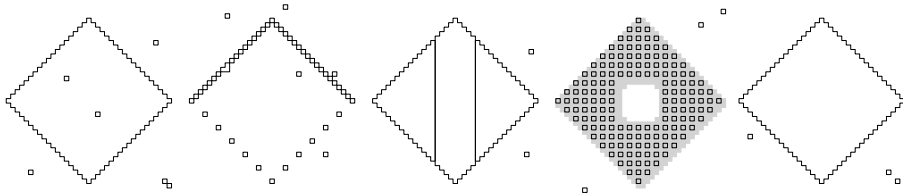


Figure 2.6. Texture—artificial image:
all shapes originally detected in the image (black) and region classified as textured (gray).

⁵ Definition according to [206]. A slightly different definition is also commonly used (see, e. g., [106]), however, for erosion by a disc centered at the origin, both versions are equivalent.

2. Extraction of Shapes

Figures 2.7 and 2.8 show the result of the texture detection for real-world trademark images from the *UK trademarks* set (see Section 1.6.2.2). Figure 2.7 (a) shows an image where the textured regions form shapes that are not represented by any other detected shape, such that replacing the texture by the region it forms reveals essential shapes. Figure 2.7 (b) shows an image where the textured regions just fill the interior of already detected shapes, however, replacing the texture significantly reduces the complexity of the image description.

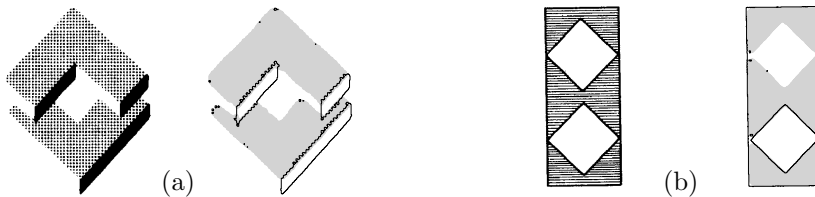


Figure 2.7. Texture—trademark images I:
original images plus all shapes not classified as texture (black)
and regions classified as textured (gray).

Figure 2.8 (a) shows an image where filigree structures inside a textured region—although easily perceptible—are also classified as being texture. Figure 2.8 (b) shows an image where a group of small shapes may be perceived as forming a shape, however is not classified as being texture.

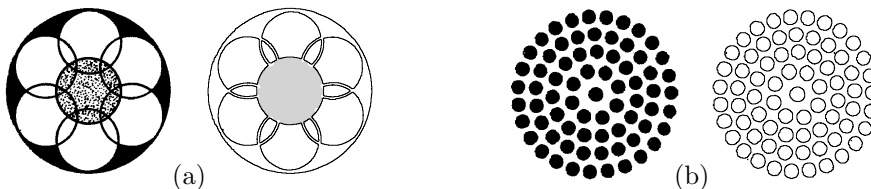


Figure 2.8. Texture—trademark images II (limitations):
original images plus all shapes not classified as texture (black)
and regions classified as textured (gray).

Applying the algorithms to the *UK trademarks* set, textured regions have been identified in 2174 (20.2%) of the 10745 image files. Averaged over all files, 313 shapes per file—which is 76.4% of the total number of shapes originally detected in the images—have been removed, but on the other hand only 0.54 shapes per file have been added. That means, that the complexity of the image descriptions is remarkably reduced.

2.3.2. Broken Lines

When an image depicts line shapes or the outlines of region shapes, in raster graphics these lines typically correspond to chains of pixels having the same color. In the vectorization stage such a chain becomes a set of thin region shapes some of which may even correspond to single pixels. Moreover, the elements of such a set need not necessarily be connected, especially in case of dashed or dotted lines. However, in order to get a representation that conforms with perception, these chains should be re-connected and the sets of thin regions should be replaced by the lines they correspond to.

Related Work If the line shapes are restricted to come from a limited set of possible curve types, e.g., straight line segments or circular arcs, these primitives can be detected based on the corresponding pixels using recognition techniques such as the *generalized Hough transform* [115, 68] or alignment based on the *random sample consensus* [84] (see Section 3.2.1 for a detailed description). However, in the given context the line shapes may not be restricted to such a limited set of primitives.

In the field of digitizing technical drawings, the same problem arises: lines have to be detected although they might be depicted as dashed or dotted lines etc. Algorithms have been proposed for straight lines (see, e.g., [131]) as well as for arbitrary curved lines (see e.g., [228], [65], and [239]), but these approaches make very restrictive assumptions about the line types: line thickness, actual lengths of dashes and gaps, or repetition patterns (see also [137]). Since the line types in technical drawings have been standardized (cf. [123]), exploiting knowledge about the rules may be quite beneficial for recognition, however, in the given context there are no such rules.

A problem that is closely related, is the so called *curve reconstruction problem*: Given a set of points sampled from a curve, connect them according to their adjacencies on the curve (see, e.g., [63]). Whether a curve may correctly be reconstructed by a given algorithm, of course depends on the sample at hand. For a curve C , let M_C be the *medial axis* (as introduced in [31]), and for a point $p \in C$ let the local feature size $lfs(p) := \min_{m \in M_C} \{\|p - m\|\}$ be the minimum distance of p to a point of the medial axis. A sample set $S \subset C$ is called ε -sample of C , if every point $p \in C$ is within distance $\varepsilon \cdot lfs(p)$ of a sample point $s \in S$ (definition according to [12]).

Approaches that do not make use of information about the direction of the curve like, e.g., using minimum spanning trees [82], using α -shapes (as defined in [76]) [26], using the so called *crust* (defined based on the Voronoi diagram and the Delaunay triangulation) [12], and using β -skeletons (as defined in [135]) [12],

2. Extraction of Shapes

are rather restrictive with respect to the sampling conditions. The latter two approaches have been shown to work correctly if the input is a 0.252-sample or a 0.297-sample respectively.

By exploiting knowledge about the direction of edges that have already been identified as belonging to the reconstruction of the curve, the restrictions on the sample can be relieved considerably. In [62] the nearest neighbor graph of the sample points is constructed and used as basis for the reconstruction. Any point b with only one incident edge $\{a, b\}$, is then connected to the nearest point c with the property that the angle spanned by \overline{ab} and \overline{bc} is greater than $\pi \cdot 1/2$ and smaller than $\pi \cdot 3/2$. This algorithm was shown to work correctly if the input is a 0.4-sample [147]. Reducing the range of allowed angles to $[\pi - 0.97, \pi + 0.97]$ even results in an algorithm capable of reconstructing curves from 0.48-samples [147].

An algorithm that is capable of reconstructing also curves having corners or intersections was presented in [146, 147]: Starting from the shortest edge, the curve is traced by iteratively searching the next sample point based on the direction and the endpoint of the edge previously added to the reconstruction. The sensitivity to the direction is achieved by using a so called *probe* which is a (typically symmetric and convex) shape with designated reference point and reference direction. Let $\{a, b\}$ be the last edge added to the reconstruction and let b be its free endpoint. The probe is aligned such that its reference point coincides with b and that its reference direction coincides with the direction from a to b . Beginning with scaling factor 0, the probe is inflated until it touches a point c from the (remaining) sample (see Figure 2.9 for an illustration). The edge $\{b, c\}$ is then added to the sample and used for the next iteration.

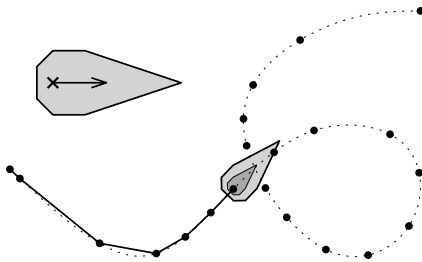


Figure 2.9. Curve reconstruction using probe: example of a polygonal probe (gray), plus self intersecting curve (dotted) with sample that can be used for reconstruction using the given probe.

The approaches based on detecting line primitives work even if the input is noisy, but they cannot be used to reconstruct arbitrary line shapes. The algorithms for curve reconstruction, on the other hand, usually require the input to be sampled from the curves only. Since positional errors of the sample points correspond to deformations of the curves, the algorithms are supposed to be robust with respect to such errors. However, algorithms applied in the given context have to be able also to deal with additional spurious points in the input. In [63] this aspect was also considered, but unfortunately the approach presented there, again waives any information about the direction of the curves.

Idea of the Proposed Approach The proposed approach extends the idea of exploiting knowledge about the direction of the curve as done in [146], however, also other features as, e. g., *good continuation* are exploited. Furthermore not only a single curve is expanded greedily by locally searching for the next edge, but all candidates are considered simultaneously.

Since the small or thin regions are supposed to represent line shapes—one dimensional objects—re-connecting the pieces is not done based on the outlines of the regions, but based on points of the skeletons. Due to discretization artefacts these points need not correspond to smooth curves, and due to noise pixels or other small shapes in the image the points need not all contribute to line shapes. In [63], curve reconstruction is characterized as “*connecting dots with good reason*”. For the proposed approach, the reasons include:

- proximity of the points
- analogy of the directions of a candidate edge and already reconstructed parts
- “goodness” of form of the resulting reconstruction

A set of polylines is generated from the points by iteratively selecting the pair of points with best priority, connecting them by an edge and updating the priorities for the other points.

2. Extraction of Shapes

Determining Suitable Sample Points—the Max- L_∞ -Skeleton For small and thin region shapes that originate from the depiction of a curve, the points used to represent them ideally should be points from the original curve. Given a polygon, skeletons like the *medial axis* as introduced in [31], or the straight line skeleton as introduced in [4] also reach for the boundary of the polygon, namely for the convex vertices. Especially for polygons originating from raster graphics this may result in skeletons that have many almost redundant edges (see Figure 2.10 (a) and (b) for examples).

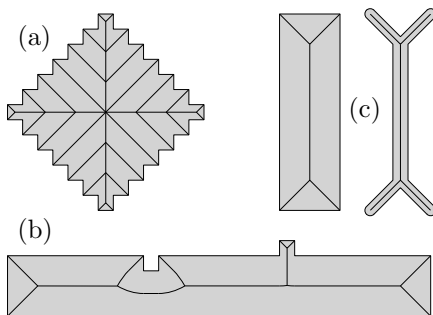


Figure 2.10. Undesirable properties of the medial axis: (a) rectilinear shape that is essentially a square, plus its medial axis with almost redundant edges, (b) rectilinear shape that is essentially a thin rectangle, but with a small indentation and a small protrusion, plus its medial axis, (c) two different shapes having the same medial axis.

In the context of shape retrieval, the relevant features of such a skeleton are usually obtained by *skeleton pruning* (see [18] for an overview). In the given context, however, not a representation capturing the characteristics of the shape, but a representation that facilitates finding possible links to other shapes is needed—for the left shape in Figure 2.10 (c), possible links are probably the upper and the lower side of the rectangle, but not the corners; for the right shape, however, possible links are probably the ends of the protrusions.

The input is supposed to consist of small or thin rectilinear polygons with integer coordinates. The output should be a set of points simple to compute, indicating the principal course of the shape and its protrusions, but not overemphasizing every convex vertex. Although the medial axis in its original form is unsuitable, the basic concept may be used in a slightly modified way. The medial axis of a region shape R can be defined as the locus of the centers of the (Euclidean) circles that are completely contained in R and that are maximal in the sense that each such circle is not a proper subset of any other circle completely contained in R . Having in mind the rectilinear nature of the input, and the demand to suppress skeleton parts reaching for nonrelevant corners of the polygon, the points used here are the centers of axis aligned

squares (L_∞ -circles) that are completely contained in the polygon and being maximal in the sense that each such square is not a proper subset of any other axis aligned square lying completely inside the polygon.

Since a set of discrete skeleton points is needed, and since the input polygon has integer coordinates, the squares are restricted also to have integer coordinates. Furthermore, in order to eliminate points due to insignificant protrusions, the longest consecutive part of the square's boundary not touching the polygon's boundary is restricted to be smaller than $1/2$ times the perimeter of the square. This implies that such a square touches the polygon in at least 2 points on opposite sides, but moreover, ensures that inconspicuous protrusions do not contribute to the skeleton. In order to also eliminate protrusions due to single pixels (for which the inscribed square is automatically touched at $3/4$ of its perimeter), squares of sidelength 1—as a special case—for contributing to the skeleton, are demanded to be neighbored by another lattice square of sidelength 1 if the polygon is larger than a single pixel. Figure 2.11 shows an illustrative example of a rectilinear lattice polygon and its max- L_∞ -skeleton.

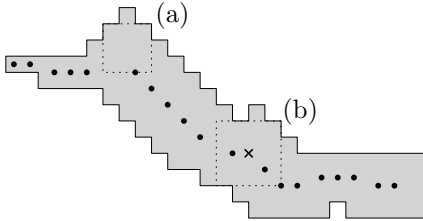


Figure 2.11. Definition of the max- L_∞ -skeleton: polygon with its skeleton points plus two maximal axis aligned lattice squares where the length of the longest consecutive part not touching the polygon is (a) $7/12 > 0.5$, and (b) $6/16 < 0.5$ times the perimeter.

For a rectilinear lattice polygon, the max- L_∞ -skeleton can easily be computed starting from a convex vertex v . A square of sidelength 1 aligned to the corner in v is inflated until it is maximal. Based on the information about the parts of the squares boundary that do not touch the polygons boundary (and correspond to regions that have not been explored so far), new maximal squares or new inflatable squares are generated and processed recursively.

Each shape for which the maximum sidelength of a square corresponding to a point of its max- L_∞ -skeleton is sufficiently small, does only consist of small or thin parts. These shapes, therefore, are considered as potentially being part of the depiction of a line shape. Shapes for which the value is large, on the other hand, are classified as region shapes.

2. Extraction of Shapes

Connecting the Points Given the set $S = \{p_1, \dots, p_n\}$ of skeleton points of thin shapes, the idea is to connect pairs of points from S according to their likeliness of being consecutive points on some curve. A set of polygonal chains \mathcal{P} is grown by iteratively determining the pair of chain endpoints with the overall highest likeliness and—if this likeliness is sufficiently high—connecting these two endpoints. The initial set $\mathcal{P}_0 = \{(p_1), \dots, (p_2)\}$ consists of polygonal chains with only one point. The basic framework therefore is the same as in Kruskal’s algorithm for computing the minimum spanning tree of a graph [139], however, here the “lengths” of the edges may change from iteration to iteration.

The likeliness of two points p_i and p_j being consecutive points on some curve is assumed to depend on the following features:

distance The Euclidean distance $\|p_i - p_j\|$. The closer the points, the higher the likeliness of being consecutive points on some curve.

distribution of nearest neighbors For every point the distribution of points in its neighborhood is analyzed. For a point that actually does originate from the depiction of a line shape, there are probably also other points from that line shape contained in the neighborhood, situated in a rather thin corridor containing the point itself. The thinner the corridor, the higher the likeliness that a point is actually part of a curve

direction The direction of the edge between points p_i and p_j in relation to the courses of the hitherto constructed chains that p_i and p_j belong to. Due to the discrete nature of the original polygonal shape and its skeleton, consecutive edges of a correct reconstruction might pretty well form a right angle. The direction of an edge, therefore, is not rated on the direction of a single predecesing edge only, but on a larger part of the reconstruction.

goodness of form The goodness of form of the resulting chain. Human perception tends to prefer simple figures over complex ones (see Section 1.2). Edges leading to curves of low visual complexity therefore are preferred over edges leading to erratic structures.

coverage The density of points on the resulting chain or more formally, the ratio of the chain’s parts covered by shape (in contrast to the gaps). The more of the line is actually depicted, the longer the gap between two parts might be. In dashed lines, e. g., the gaps may surely be larger than in dotted lines without destroying the perception of a line—especially in the presence of noise.

A more detailed description of these features is given in the Appendix (Section B.1). The list of features used is not claimed to be a complete list of relevant features and, moreover, the influences of the features on the perception of lines have not been studied quantitatively. However, the idea is to derive a value of ‘*weighted distance*’ d_w (which serves as an estimate of the inverse of the likeliness of being consecutive points on some curve), and to use this value in the Kruskal-like framework for computing polygonal reconstructions. The basic structure is outlined in Algorithm 2.1.

Algorithm 2.1:

```

 $\mathcal{P} \leftarrow$  init chains
 $\mathcal{E} \leftarrow$  create (relevant) weighted edges
while minimum edge weight < threshold do
     $(p_i, p_j) \leftarrow$  extract minimum
    merge chains at  $p_i$  and  $p_j$ ; update  $\mathcal{P}$ 
    if  $p_i$  not singleton then
        | remove edges containing  $p_i$ 
    end
    if  $p_j$  not singleton then
        | remove edges containing  $p_j$ 
    end
    update weights of edges containing opposite end of  $p_i$ 
    update weights of edges containing opposite end of  $p_j$ 
end
return sufficiently long elements of  $\mathcal{P}$ 

```

Starting from the set $\mathcal{P}_0 = \{(p_1), \dots, (p_2)\}$ of trivial chains, the weighted distance for every (relevant) pair of end points is computed and used as priority to store the pair in a priority queue. While the minimum priority is sufficiently small, the corresponding pair is extracted from the queue and connected by an edge. Let (p_{i_1}, p_{j_1}) be this pair and let $P_i = (p_{i_1}, \dots, p_{i_k})$ and $P_j = (p_{j_1}, \dots, p_{j_l})$ be the chains the two points belong to (pairs containing p_{i_k} or p_{j_l} are handled accordingly). If $P_i = P_j$, meaning that p_{i_1} and p_{j_1} are opposite endpoints of the same chain, then the new edge closes this chain and since p_{i_1} and p_{j_1} cannot be connected to further points, all pairs containing either p_{i_1} or p_{j_1} are invalidated. If p_{i_1} and p_{j_1} are the endpoints of different chains, then—provided that the respective chain did not consist of a single point—all pairs containing either p_{i_1} or p_{j_1} are invalidated, and the priorities (weighted lengths) of all relevant pairs containing one of the opposite endpoints p_{i_k} or p_{j_l} are updated.

2. Extraction of Shapes

In the case that the deviation of the weighted distance d_w from the Euclidean distance d can be bounded such that $c_l \cdot d < d_w < c_u \cdot d$ with $c_l > 0$ and $c := c_u/c_l$ being constant, the number of point pairs that have to be considered can be reduced using the Delaunay triangulation (cf. [59]) of the points: For $d_{w,min}$ being the current minimum weighted distance between two chain end points, only point pairs with Euclidean distance smaller than $c \cdot d_{w,min}$ have to be considered.

Experimental Results The approach has been tested using an implementation with the set of parameters described in the Appendix (Section B.1). Figure 2.12 shows the result of the line reconstruction for the exemplary image from Figure 2.3.

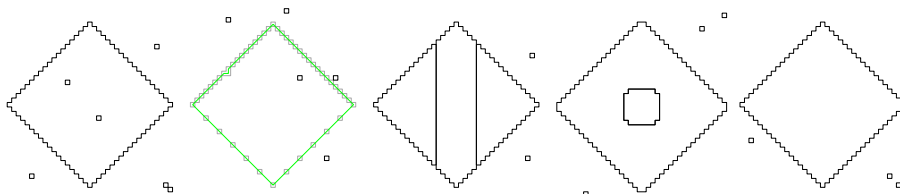


Figure 2.12. Line reconstruction—artificial image I: line shapes that have been detected (green) and small regions that can be replaced (gray).

Figure 2.13 shows the result of the line reconstruction for a noisy image depicting two curves⁶ in a scattered way. Except for one out of five crossings the courses of both curves are correctly detected.

Figure 2.14 shows the result of the line reconstruction for real-world trademark images from the *UK trademarks* set (see Section 1.6.2.2). The left part shows an 832×465 pixel image depicting thick dashed lines, the right part shows an 346×255 pixel detail of an 912×1269 pixel image depicting very thin lines by discontinuous parts.

⁶ Straight line: $(t, 1 - t/3)$, curved line: $(t, \sin(t^2))$ for $t \in [0, \sqrt{4.5 \cdot \pi}]$.

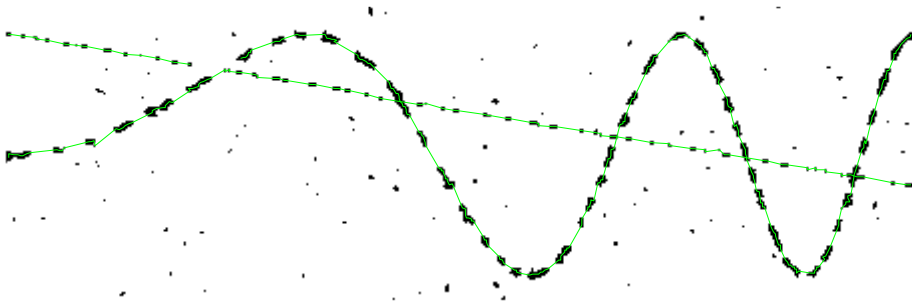


Figure 2.13. Line reconstruction—artificial image II: noisy image (black and white) depicting two curves, plus line shapes that have been detected (green).

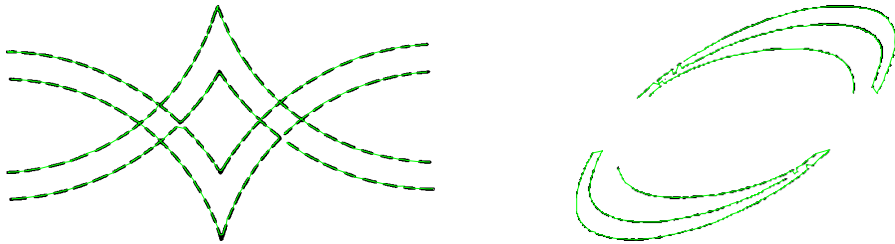


Figure 2.14. Line reconstruction—trademark images: two original images (black and white) plus line shapes that have been detected (green).

Applying the algorithms to the *UK trademarks* set, line shapes have been reconstructed in 3 439 (32.0%) of the 10 745 image files. Averaged over all files, 14.3 shapes per file—which is 3.5% of the total number of shapes originally detected in the images—have been removed, while only 2.6 shapes per file have been added. On the other hand, averaged over the files where line shapes have actually been reconstructed, these line shapes constitute 32.4% of the shapes in the final image representations. That indicates that for these images line shapes play an important role.

2.4. Simplification

The data extracted from raster graphics usually contain a huge amount of information that is actually not perceived in the images, e. g., shapes originating from noise, or almost redundant vertices in the polygonal chains. The goal is, to get representations that, on the one hand, have low complexity and, on the other hand, capture the essence of the perceived shapes.

2.4.1. Polyline Simplification

The polylines generated in the vectorization step consist of axis aligned line segments of length 1, since the vertices are just the corners of the pixels. For further processing, the complexity (number of vertices) of these polylines has to be reduced. All redundant vertices can be removed in time linear to the total number of vertices by simply traversing the polylines and testing every triple of consecutive vertices for collinearity. However, inclined edges depicted in raster graphics result in stairlike lines (so called *jaggies*), and noise may also cause additional vertices. The detected polylines therefore have to be simplified in a lossy way.

2.4.1.1. Basics

Numerous methods have been developed for the task of polyline simplification—replacing a polyline P by a polyline Q , such that the number of vertices of Q is smaller than the number of vertices of P and that Q is a ‘good approximation’ of P (see [214] and [161] for an overview). The criteria for Q being a ‘good approximation’, of course, depend on the application at hand—in [160] even 30 possible criteria for polyline simplification only in the context of cartography are listed.

Basic Quality Criteria Cartography is a typical field of application for polyline simplification algorithms. The data about, e. g., frontiers, river courses, or isohypsometric lines as originating from land surveying should be as detailed as possible. However, for convenient visualization and efficient rendering at a given scale, the polylines representing this data have to be simplified, preferably in a way such that only the details visible at that scale are kept. In this context the reference for evaluating the quality of an approximation Q surely is the input polyline P . Therefore in many cases ‘good approximation’ means—although virtually never stated this way—that the weak Fréchet distance $d_{wF}(P, Q)$ is small.

Justifiably, a major part of the research is focused at this criterion. In addition, also topological constraints have been considered such as, e.g., that non self-intersecting polylines are always approximated by polylines that are also non self-intersecting [246], or for sets of polygonal chains (like topographic contour lines) that the combinatorial structure of the induced planar subdivision is preserved [25, 77].

However, when an artificially designed shape as given by a polyline P^* is depicted in an image, the polyline P detected in this image may contain rasterization artefacts, noise, and artefacts due to the re-vectorization. In this case, the reference for evaluating the quality of an approximation Q should be the original polyline P^* and not the input P of the polyline simplification.

Quality Ratings for Filtering Techniques A polyline simplification method may either allow arbitrary points as vertices for the new polyline, or it may only allow subsequences of the vertices of the input polyline. The latter are called *filtering* techniques and for these methods the quality of the approximation is often rated based on the following considerations. Let (p_1, \dots, p_n) be the vertices of the input polyline P and let (a_1, \dots, a_k) with $a_i < a_j$ for $i < j$ be the indices such that $(p_{a_1}, \dots, p_{a_k})$ are the vertices of the approximation Q . A part $c_i = (p_{a_i}, p_{a_i+1}, \dots, p_{a_{i+1}-1}, p_{a_{i+1}})$ of P can be associated to the line segment $s_i = \overline{p_{a_i} p_{a_{i+1}}}$ of Q (see Figure 2.15 for an example).

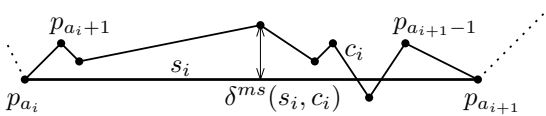


Figure 2.15. Definition of the local error criterion δ^{ms} .

Using a local error criterion δ that measures the quality of a line segment s with respect to a part c , the overall quality of an approximation Q is often rated as $\Delta(Q, P) := \max_{1 \leq i < k} \{\delta(s_i, c_i)\}$. For a line segment s and its associated part c , one of the most commonly used local error criteria is the maximum distance of a vertex of c to s , but also other local error criteria have been considered (see, e.g., [120] and [38]), and $\Delta(Q, P)$ need not necessarily be defined by the maximum. In the following the overall error criterion defined as the maximum of the local errors will be referred to as Δ^m and the local error criterion defined as the maximum distance of a vertex to the line segment will be referred to as δ^{ms} .

2. Extraction of Shapes

With s being a single straight line segment, the local error criterion δ^{ms} actually corresponds to the Hausdorff distance $d_H(s, c)$ and to the weak Fréchet distance $d_{wF}(s, c)$. However, in general a single line segment might not be the only one used to approximate the associated part. The (weak) Fréchet distance of an approximation to the input polyline, therefore, might be smaller than the maximum of the (weak) Fréchet distances of all the line segments to their associated parts. So optimizing with respect to such a local error criterion does not necessarily yield a globally optimal result (see Figure 2.16 for an example). Therefore, a formulation like “*simplification under the Fréchet error measure*” as used, e. g., in [3] must not be mistaken when it refers to the error criterion Δ^m derived from a local error criterion δ , instead of referring to the Fréchet distance d_F which is defined on the complete polylines.

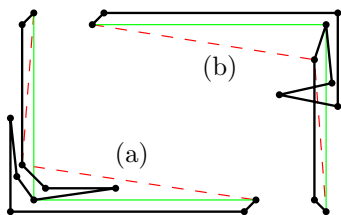


Figure 2.16. Different optima: polylines (black) for which approximations optimal with respect to a local error criterion (dashed red), differ from the corresponding globally optimal approximation (thin green) (a) Hausdorff / weak Fréchet distance, (b) Fréchet distance.

Local Optimal Filtering Algorithms Given an input polyline P and an error criterion Δ , there are two possible ways of formulating an optimization problem:

min-# approximation For a fixed bound Δ_{max} on the error, find the approximation Q with $\Delta(Q, P) \leq \Delta_{max}$ minimizing the number of vertices.

min- Δ approximation For a fixed number k of vertices, find the approximation Q with $|Q| \leq k$ minimizing $\Delta(Q, P)$.⁷

If Δ is defined using a local error criterion δ (combined by an operation that is associative), the min-# approximation problem is often formulated in terms of computing the shortest path in some directed acyclic graph and can be solved using dynamic programming [120]. The vertices of the graph correspond to the vertices of P and an edge (p_i, p_j) of the graph corresponds to the straight line segment $\overline{p_i, p_j}$. Depending on the definitions of Δ , of δ , and on the value of Δ_{max} , the graph might be weighted and might be incomplete (in the sense that it need not contain every possible edge (p_i, p_j) with $i < j$).

⁷ In the literature this is often referred to as *min- ε approximation*.

Since a shortest path in a directed acyclic graph $G = (V, E)$ can be computed in time $O(|E|)$, the total running time of an algorithm solving a min-# approximation problem for an open polyline is determined by the time needed to construct the graph (or to compute the weights). For the local error criterion δ^{ms} , e.g., the trivial time bound of $O(|P|^3)$ can be reduced to $O(|P|^2)$ [44]. A closed polyline might be split at $|P|$ different vertices, however, a min-# approximation can be computed solving an all-pairs-shortest-paths problem on a graph with $O(|P|)$ vertices.

If Δ is defined as the maximum of some local error criterion δ for an edge of the approximation, the resulting value really has to occur on some edge. A min- Δ approximation may then be computed by determining the set of all possible values that δ assumes, sorting them and performing a binary search on the values with solving the corresponding min-# approximation problem in every step [120]. This way, for the local error criterion δ^{ms} a min- Δ approximation of an open polyline can be computed in time $O(|P|^2 \cdot \log(|P|))$.

Local Suboptimal Filtering Algorithms In [161] the approaches for polyline simplification have been divided into 5 categories:

1. **independent point algorithms** approaches that do not take account of the topology or geometry of the polyline, e.g., selecting every $(n/k)^{\text{th}}$ vertex or selecting k random vertices,
2. **local processing routines** approaches that take account of characteristics of immediate neighboring vertices, e.g., distance or angular change,
3. **constrained extended local processing routines** approaches that take account of sections of the polyline constrained by, e.g., distance or number of vertices,
4. **unconstrained extended local processing routines** approaches that take account of sections of the polyline constrained by geomorphological complexity,
5. **global routines** approaches that take account of the entire line.

Algorithms of categories 1–3 are not appropriate in the context of extracting shapes from figurative images. It is quite easy to generate polylines and also to think of real-world examples where they produce poor results.

2. Extraction of Shapes

Pavlidis and Horowitz [179] proposed to use the general idea of split-and-merge procedures also for polyline simplification. This idea will be briefly presented in the following as it allows to create algorithms of category 4 and 5 that also satisfy the need for adaptive simplification (see below). In terms of filtering algorithms, a general split-and-merge approach works as follows: Given an input polyline that is subdivided into parts, each of which is approximated by a line segment, in the split-phase all parts which are not approximated with appropriate quality get split into two parts. In the merge-phase consecutive parts for which the union would be properly approximated by a single line segment get merged. These two phases can also be iterated until no further progress is achieved. Algorithms of this type may differ in the used quality criterion and in the way the split vertex is selected.

One of the most commonly used methods for polyline simplification is the *Douglas-Peucker* algorithm as presented in [66]. Due to its broadly acknowledged good performance, the Douglas-Peucker algorithm will also be used as a reference in the present work. The algorithm fits into the split-and-merge scheme, but waives the merge-phase. Given an open polyline $P = (p_1, \dots, p_n)$ and an error bound Δ_{max} the complete polyline is initially approximated by the single line segment $\overline{p_1 p_n}$. As long as the approximation contains a segment $s_i = \overline{p_l p_r}$ with associated part c_i such that $\delta^{ms}(s_i, c_i) > \Delta_{max}$, the chain c_i is split at the vertex p_j that maximizes the distance to s_i and the approximating segment s_i is replaced by $\overline{p_l p_j}$ and $\overline{p_j p_r}$. A straightforward implementation has a worst case running time quadratic in the number of vertices of P , but there is also an improved version that achieves a worst case running time of $O(|P| \cdot \log(|P|))$ by using information about the convex hulls of the parts in order to find the split vertices more efficiently [109].

Adaptive Approximation Since figurative images are artificially designed, many depicted shapes are bounded by long straight line segments or smooth curves. Even when these boundaries get disturbed due to noisy representations, the original shapes are usually unfailingly recognized by humans. That means, that for approximating a polyline representing such a long straight line segment, a relatively large error bound might be acceptable. However, in curved parts of the shape, the same error bound could be too large so that the perception of the shape would be changed. The error bound therefore should be adaptively chosen, depending on the course of the polyline.

Adaptive approximation of polylines was already considered in a different context, namely the simplification of hand drawings. As opposed to usual digitalization artefacts, in hand drawings the deviation from a straight line segment often increases with increasing length of the segment. Therefore, it was suggested to consider errors normalized with respect to the length of

the segment [178]. However, in the given context the same deviation from a straight line segment might be perceived very differently, depending on the overall *goodness* of the polyline to approximate. Figure 2.17 shows an example from a real-world trademark (a), where a very ragged part of the boundary would ideally be approximated by a single straight line segment. For a shape with a very straight border (b) on the other hand, the same deviation—absolute as well as relative—from a straight line would be perceived as feature, rather than noise.

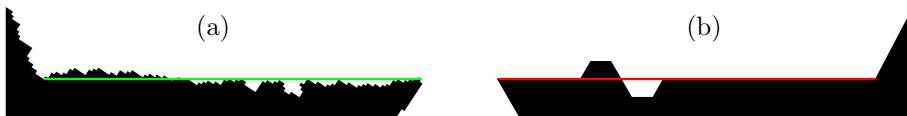


Figure 2.17. Adaptive simplification: (a) rotated detail of a real trademark image where the green line would be a reasonable approximation, (b) constructed shape where the red line would *not* be an acceptable approximation although the maximum deviation δ^{ms} is not larger.

A different way to implement adaptive simplification is to break the given polyline into pieces depending on the curvature⁸, and to independently choose an appropriate level of simplification for each part [91]. However, figurative images also often depict shapes where the curvature does not change abruptly, which makes finding a useful partition based on curvature difficult.

Split-and-merge approaches such as the Douglas-Peucker algorithm on the other hand offer the opportunity to use the information about the raggedness of a part for deciding whether the part needs to be split at all. That means that no fixed partition of the polyline has to be applied.

Approximation with Respect to the Original Polyline As mentioned above, in the given context it is not the goal to find a polyline approximating the input polyline P , but approximating the original polyline P^* . Assuming, that P approximates P^* within some error bound Δ_1 , any polyline Q that approximates P within some error bound Δ_2 obviously guarantees to approximate P^* within $\Delta_1 + \Delta_2$, however, the actual error might be even smaller.

⁸ After smoothing, e. g., using a Gaussian filter.

2. Extraction of Shapes

The Douglas-Peucker algorithm splits the parts at the points that have maximal distance from the associated line segment. However, such a vertex need not be perceived as being conspicuous. Especially when parts of the polyline are almost parallel to the associated line segment, the vertex chosen by the Douglas-Peucker algorithm might be a virtually redundant one which does not lie near any vertex of the original polyline P^* (see Figure 2.18 for an example). Given an input polyline P and a polyline Q that deviates from P by Δ_2 , there is no indication whether the deviation is caused by noise which is eliminated or whether the deviation is caused by a change of the polyline's appearance. However, evaluating the quality of the approximation, using the polyline P^* from which P originated from, reduces this uncertainty.

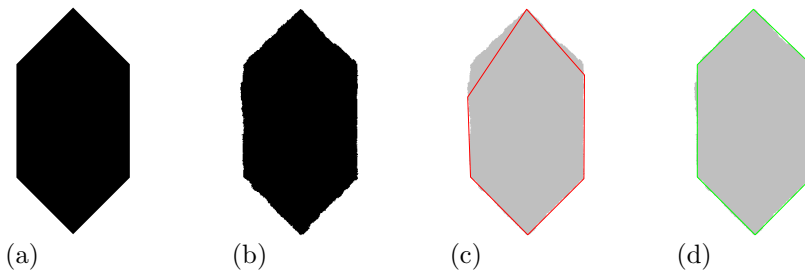


Figure 2.18. Undesirable behavior of Douglas-Peucker algorithm:
(a) original shape, (b) distorted shape, (c) outline of distorted shape simplified using Douglas-Peucker algorithm, (d) outline of distorted shape simplified using coarsening algorithm as described in Section 2.4.1.2.

Using some kind of original polyline instead of the input polyline for evaluating the quality of an approximation was already considered in [102]. However, in that work the deviations from the original polyline were modeled as resulting from white noise, which is surely not an appropriate assumption for polylines originating from raster graphics.

In the present work three types of errors—namely deformation, blur, and individual erroneous pixels along the boundary—were considered to experimentally evaluate the performance of different simplification strategies. Given a bi-level black-and-white raster graphic depicting a simple shape, deformation was emulated by walking along the countour of the shape and performing a local dilation or erosion with a circle of variable diameter. Blurring of the shapes was realized applying a simple binary low-pass filter on the image. The addition and deletion of individual pixels along the boundary was realized in a straightforward way. Figure 2.18 (b) shows a 6-gon distorted using all three kinds of errors.

2.4.1.2. Coarsening Simplification

Idea of the Proposed Approach By recursively splitting the polyline into parts, the Douglas-Peucker algorithm selects vertices where the polyline should *not* be simplified and the actual simplification is achieved by disregarding all the vertices inbetween. A strategy that is opposed to that one, would be to actively determine vertices where the polyline should be simplified and remove them. While the Douglas-Peucker algorithm is a split-and-merge approach that waives the merge phase, the proposed algorithm is a split-and-merge approach that waives the split phase. For deciding whether two neighboring edges should be merged, again the local error criterion δ^{ms} is used. If the value is below a given threshold, the single line segment connecting the unshared endpoints of the two edges is supposed to be a good approximation for the whole part.

Algorithm The basic structure is outlined in Algorithm 2.2.

Algorithm 2.2:

```

 $\mathcal{E} \leftarrow$  create pairs of neighboring edges
while minimum error of an edge pair < threshold do
     $(e_l, e_r) \leftarrow$  extract minimum from  $\mathcal{E}$ 
     $e \leftarrow$  merge  $e_l$  and  $e_r$ 
    create pair containing left neighbor of  $e_l$  and  $e$ , add to  $\mathcal{E}$ 
    create pair containing  $e$  and right neighbor of  $e_r$ , add to  $\mathcal{E}$ 
end
return resulting chain of edges

```

Let the input polyline P be given by the sequence of its vertices (p_1, \dots, p_n) . For every pair of neighboring edges $(\overline{p_l p_m}, \overline{p_m p_r})$ (initially $l + 1 = m = r - 1$) the error $\delta_{l,r} := \delta^{ms}(\overline{p_l p_r}, (p_l, \dots, p_r))$ is computed as the maximum distance of a vertex from (p_l, \dots, p_r) to the segment $\overline{p_l p_r}$. Every such pair $(\overline{p_l p_m}, \overline{p_m p_r})$ is added to a priority queue using its error $\delta_{l,r}$ as priority.

While the minimum of the priorities is smaller than a given error bound Δ_{max} , the pair $(\overline{p_b p_c}, \overline{p_c p_d})$ belonging to that minimum is removed from the queue and if it is still a valid pair of edges (a pair becomes invalid if its left or right edge has been merged with another edge before) it is merged to the single edge $\overline{p_b p_d}$. For its current left neighbor $\overline{p_a p_b}$ and its current right neighbor $\overline{p_d p_e}$ the errors $\delta_{a,d}$ and $\delta_{b,e}$ respectively are computed as described above, and the corresponding pairs are added to the priority queue.

Finally, the approximating polygon Q is reconstructed from the edges present after the merging phase. The whole algorithm can easily be generalized to be used with closed polylines also.

2. Extraction of Shapes

Analysis Since only a constant number of edge pairs (namely 2) is evaluated after a merge is performed and since the number of merges is bounded from above by the number of vertices of the input polyline P , the total number of edge pairs considered during the algorithm is in $O(|P|)$. Using a heap as priority queue, storing and retrieving an edge can be done in time $O(\log|P|)$. A straightforward implementation of the evaluation of the errors $\delta_{l,r}$ leads to a worst case total running time that is quadratic in the number of vertices of P . However, following the ideas of [109], information about the convex hulls of the merged parts can be used to achieve a worst case running time that is in $O(|P| \cdot \log(|P|))$.

2.4.1.3. Adaptive Coarsening Simplification

Idea of the Proposed Approach The need for adaptive simplification can easily be met by a simple modification of the coarsening algorithm: Every candidate edge $\overline{p_l p_r}$ of an approximating polyline gets a quality rating depending on its error. If this quality is higher than the combination of the ratings of the best previously existing finer chain of edges $(\overline{p_l p_{m_1}}, \dots, \overline{p_{m_h} p_r})$, the chain may get replaced by $\overline{p_l p_r}$ even if the error exceeds the primary error bound Δ_{max} . On the one hand, in this way ragged depictions of straight lines may be replaced by a single line segment. On the other hand, curvilinear parts and small features in straight regions are not destroyed, when the quality is decreasing for coarser and coarser approximations. In order not to tolerate errors that, due to their absolute values, are not perceived as resulting from noise, the merging phase is stopped when a certain threshold $\Delta_{break} > \Delta_{max}$ is exceeded.

The general structure of the coarsening algorithm can be kept. However, the merging of edges is now performed ignoring Δ_{max} as an upper bound. Edges with error smaller than Δ_{max} are taken in any case. If the error is larger, a quality rating for the single edge itself and for an alternative chain of subordinate edges is computed. These ratings are used afterwards to decide whether the merge led to a more appropriate representation or whether the finer subdivision should not have been replaced. The construction of the approximating polyline starts with the final edges and based on the quality ratings decides whether an edge is part of the approximation or whether the corresponding merge operation has to be ignored and subordinate edges are recursively taken.

Algorithm The basic structure is outlined in Algorithm 2.3.

Algorithm 2.3:

```

 $\mathcal{E} \leftarrow$  create pairs of neighboring edges
while minimum error of an edge pair  $< \Delta_{break}$  do
     $(e_l, e_r) \leftarrow$  extract minimum from  $\mathcal{E}$ 
     $e \leftarrow$  merge  $e_l$  and  $e_r$ 
    rate whether  $e$  is favorable or not
    create pair containing left neighbor of  $e$  and  $e$ , add to  $\mathcal{E}$ 
    create pair containing  $e$  and right neighbor of  $e$ , add to  $\mathcal{E}$ 
end
recursively break unfavorable edges
return resulting chain of edges

```

Let the input polyline P be given by the sequence of its vertices (p_1, \dots, p_n) . For every pair of neighboring edges $(\overline{p_l p_m}, \overline{p_m p_r})$ (initially $l + 1 = m = r - 1$) the error $\delta_{l,r} := \delta^{ms}(\overline{p_l p_r}, (p_l, \dots, p_r))$ is computed as the maximum distance of a vertex from (p_l, \dots, p_r) to the segment $\overline{p_l p_r}$. Every such pair $(\overline{p_l p_m}, \overline{p_m p_r})$ is added to a priority queue using its error $\delta_{l,r}$ as priority.

While the number of performed merge operations is smaller than $n - 2$ and the minimum priority is smaller than Δ_{break} , the pair $(\overline{p_b p_c}, \overline{p_c p_d})$ belonging to the minimum priority is removed from the queue and if it is still a valid pair of edges (a pair becomes invalid if its left or right edge has been merged with another edge before) it is merged to the single edge $\overline{p_b p_d}$. For its current left neighbor $\overline{p_a p_b}$ and its current right neighbor $\overline{p_d p_e}$ the errors $\delta_{a,d}$ and $\delta_{b,e}$ respectively are computed as described above, and the corresponding pairs are added to the priority queue.

Every new edge $\overline{p_b p_d}$ gets a reference to the two edges $\overline{p_b p_c}$ and $\overline{p_c p_d}$ that it replaces, and a value $\omega_{b,d}^b$ describing the quality of the best representation of (p_b, \dots, p_d) . The quality of the edge itself is rated based on the relative error as $\omega_{b,d}^s = \|p_b - p_d\| \cdot \varrho_{ac} - \delta_{b,d}$ with ϱ_{ac} being a parameter determining the error tolerance. The quality $\omega_{b,d}^a$ of the alternative chain is defined based on the ratings for the two subordinate edges $\overline{p_b p_c}$ and $\overline{p_c p_d}$ as the maximum of $\omega_{b,c}^b$ and $\omega_{c,d}^b$. If the quality $\omega_{b,d}^s$ of the edge exceeds the quality $\omega_{b,d}^a$ of the alternative chain, then the merge (including all subordinate merges) leads to an improved representation, the corresponding edge is marked *favorable* and $\omega_{b,d}^b$ is set to $\omega_{b,d}^s$. In the other case, the edge is not favorable and $\omega_{b,d}^b$ is set to $\omega_{b,d}^a$. Edges with error smaller than Δ_{max} are marked *favorable* by default. As an illustration, Figure 2.19 shows two polylines, their simplifications and the hierarchy of the edges considered by the adaptive coarsening algorithm.

2. Extraction of Shapes

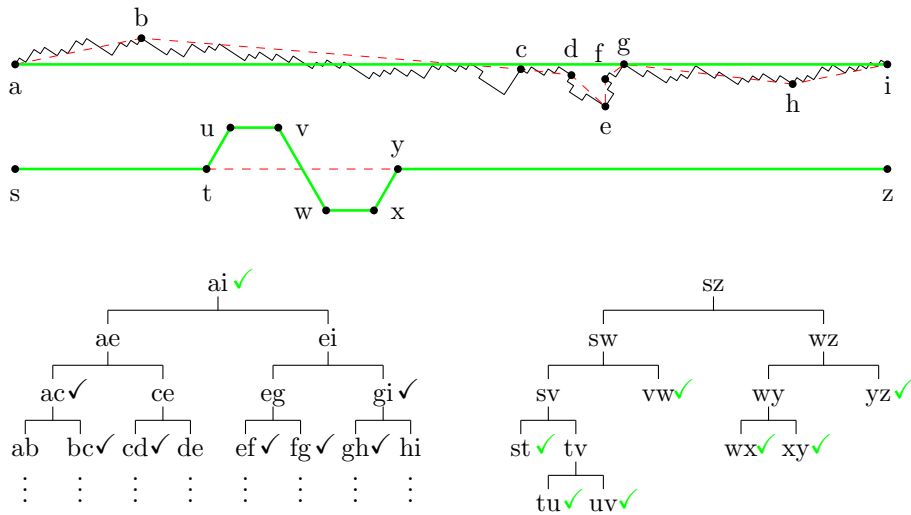


Figure 2.19. Adaptive coarsening: the two examples from Figure 2.17 with rejected edges (dashed red) and chosen edges (green), plus the corresponding tree structures on the edges, where favorable edges are marked with \checkmark .

Finally, the approximating polygon Q is reconstructed from the edges present after the merging phase. Recursively, every edge that is not *favorable* is replaced by the two edges it replaced. The whole algorithm can easily be generalized to be used with closed polylines also.

Analysis For the computation of the errors and the merging of edges, the same arguments as used in the analysis of the basic coarsening algorithm hold. Furthermore, a quality rating can be derived from the errors in constant time and the unfavorable edges form a forest which can be traversed in linear time. The total running time therefore is—just like for the basic coarsening algorithm—in $O(|P| \cdot \log(|P|))$.

2.4.1.4. Coarsening plus Corner Simplification

Idea of the Proposed Approach The depictions of shapes in figurative images (given as raster graphics) often have pointed parts where the tip is actually truncated. Reasons for the occurrence of such truncated tips include for example rounding off corners during the design of a shape or discretization in the vectorization of an image that contains antialiased or smoothed edges. A pointed part of a geometrical shape, however, is normally perceived as a perfect tip, even if it is slightly truncated. Therefore such parts may be replaced by a perfect tip (like also proposed in [71]) without changing the visual appearance of the shape. This leads to an additional approach for reducing the number of vertices of a polyline.

Let $e_1 = \overline{p_b p_d}$ and $e_2 = \overline{p_s p_t}$ be two non consecutive edges of a polyline such that their supporting lines form a pointed angle and let p be the point where both supporting lines intersect. If the distances of the endpoints p_d and p_s to p are sufficiently small compared to the lengths of e_1 and e_2 , and if all polyline vertices (p_d, \dots, p_s) lie in an ε^{cc} -neighborhood of the triangle spanned by p_d , p and p_s for a sufficiently small ε^{cc} , then the whole truncated tip (p_b, \dots, p_t) may be replaced by the pointed tip (p_b, p, p_t) . Figure 2.20 shows an example.

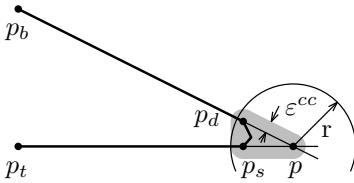


Figure 2.20. Corner simplification: polyline for which a truncated corner might be replaced by a pointed tip.

Since detecting the tips in this way requires the edges forming a tip to be long compared to the truncated part, the simplification of corners should not be performed before a simplification of the edges is carried out. On the other hand, the tip spanned by two edges $\overline{p_b p_d}$ and $\overline{p_s p_t}$ might be unfavorably replaced by a pair of edges $(\overline{p_b p'}, \overline{p' p_t})$ with p' being any vertex from (p_d, \dots, p_s) and therefore, refraining the simplification of corners until the simplification of the edges is finalized also may lead to inferior results. However, the simplification of corners can be integrated into the coarsening algorithm by searching the parts left and right of any newly formed edge e for edges e' that form a tip with e . The proposed approach allows the approximating polyline to contain vertices other than the ones from the input polyline. Therefore formally, it is not a filtering technique. On the other hand, the vertices cannot be chosen arbitrarily and in practice, most vertices will originate from the input polyline.

2. Extraction of Shapes

Algorithm The basic structure is outlined in Algorithm 2.4.

Algorithm 2.4:

```

 $\mathcal{E} \leftarrow$  create pairs of neighboring edges
while minimum error of an edge pair < threshold do
     $(e_l, e_r) \leftarrow$  extract minimum from  $\mathcal{E}$ 
     $e \leftarrow$  merge  $e_l$  and  $e_r$ 
    if  $\exists$  edge  $e_{ll}$  forming a corner left of  $e$  then
        | adapt  $e_{ll}$  and  $e$ , remove edges inbetween
    end
    if  $\exists$  edge  $e_{rr}$  forming a corner right of  $e$  then
        | adapt  $e$  and  $e_{rr}$ , remove edges inbetween
    end
    create pair containing left neighbor of  $e$  and  $e$ , add to  $\mathcal{E}$ 
    create pair containing  $e$  and right neighbor of  $e$ , add to  $\mathcal{E}$ 
end
return resulting chain of edges

```

Let the input polyline P be given by the sequence of its vertices (p_1, \dots, p_n) . For every pair of neighboring edges $(\overline{p_l p_m}, \overline{p_m p_r})$ (initially $l + 1 = m = r - 1$) the error $\delta_{l,r} := \delta^{ms}(\overline{p_l p_r}, (p_l, \dots, p_r))$ is computed as the maximum distance of a vertex from (p_l, \dots, p_r) to the segment $\overline{p_l p_r}$. Every such pair $(\overline{p_l p_m}, \overline{p_m p_r})$ is added to a priority queue using its error $\delta_{l,r}$ as priority.

While the minimum of the priorities is smaller than a given error bound Δ_{max} , the pair $(\overline{p_b p_c}, \overline{p_c p_d})$ belonging to that minimum is removed from the queue and if it is still a valid pair of edges (a pair becomes invalid if its left or right edge has been merged with another edge before) it is merged to the single edge $\overline{p_b p_d}$.

Let p_s be one of the successors of p_d in the current polyline and let p_t be the current right neighbor of p_s . Let furthermore p be the point where the supporting lines of $\overline{p_b p_d}$ and $\overline{p_s p_t}$ intersect. If the distances $\|p_d - p\|$ and $\|p_s - p\|$ are smaller than $r := \varrho^{cc} \cdot \min(\|p_b - p_d\|, \|p_s - p_t\|)$ for a predefined threshold ϱ^{cc} on the relative distance, and if all vertices (p_d, \dots, p_s) are inside the ε^{cc} -neighborhood of the triangle spanned by p_d , p and p_s for a predefined ε^{cc} , and if p_s is the last successor of p with these properties, then (p_b, \dots, p_t) is replaced by (p_b, p, p_t) . Since $r \leq \varrho^{cc} \cdot \|p_b - p_d\|$ and since the triangle spanned by p_d , p and p_s is contained in an r -ball around p , it suffices to incrementally test the successors of p_d until the distance to p_d exceeds $2 \cdot \varrho^{cc} \cdot \|p_b - p_d\| + \varepsilon^{cc}$. On the left side of the new edge $\overline{p_b p_d}$ the predecessors of p_b are processed analogously.

For the current left neighbor $\overline{p_a p_b}$ and the current right neighbor $\overline{p_d p_e}$ of the new edge $\overline{p_b p_d}$ (or the corresponding surrogates if corners were replaced) the errors $\delta_{a,d}$ and $\delta_{b,e}$ respectively are computed as described above, and the corresponding pairs are added to the priority queue.

Finally, the approximating polygon Q is reconstructed from the edges present after the merging phase. The whole algorithm can easily be generalized to be used with closed polylines also.

Analysis The number of possible successors (and predecessors respectively) for a new edge is of course in $O(|P|)$ and each such vertex can be checked in time $O(|P|)$. With the linear bound on the number of merges that may be performed, the overall running time therefore is in $O(|P|^3)$.

It is indeed possible to construct polylines such that checking m successors for a special edge takes time $\Omega(m^2)$ and no vertex gets replaced at all. However, due to the constraints on the length of the gaps relative to the length of the edges, at least for $\rho^{cc} < 1/3$ and $\varepsilon^{cc} = 0$ such a polyline requires to contain edges such that their length ratio is exponential in m , which is not possible for polylines originating from raster graphics of polynomial size.

In experiments with real-world data⁹ for the newly merged edges, the average number of other edges that were considered (successors plus predecessors) was 2.6. The average number of potential tips that had been examined, however, was only 0.43. Furthermore, for the examined tips, the average number of vertices that had to be tested was only 0.049 which means that most of the potential tips would replace a single edge.

2.4.1.5. Adaptive Coarsening plus Corner Simplification

The ideas from the *adaptive coarsening simplification* and from the *coarsening plus corner simplification* may also be realized in a single approach. As the combination of the two algorithms is very straight forward, it will not be described in detail here. The only thing that should be mentioned is that since un-favorable edges either get rejected or replaced, corners only have to be simplified for edges that are favorable.

⁹ For the *UK trademarks* set from each image the polylines were extracted, the longest polyline was chosen, and redundant vertices were eliminated before applying the coarsening plus corner simplification.

2. Extraction of Shapes

The basic structure is outlined in Algorithm 2.5.

Algorithm 2.5:

```
 $\mathcal{E} \leftarrow$  create pairs of neighboring edges
while minimum error of an edge pair  $< \Delta_{break}$  do
     $(e_l, e_r) \leftarrow$  extract minimum from  $\mathcal{E}$ 
     $e \leftarrow$  merge  $e_l$  and  $e_r$ 
    rate whether  $e$  is favorable or not
    if  $e$  is favorable then
        if  $\exists$  favorable edge  $e_{ll}$  forming a corner left of  $e$  then
            adapt  $e_{ll}$  and  $e$ , remove edges inbetween
        end
        if  $\exists$  favorable edge  $e_{rr}$  forming a corner right of  $e$  then
            adapt  $e$  and  $e_{rr}$ , remove edges inbetween
        end
    end
    create pair containing left neighbor of  $e$  and  $e$ , add to  $\mathcal{E}$ 
    create pair containing  $e$  and right neighbor of  $e$ , add to  $\mathcal{E}$ 
end
recursively break unfavorable edges
return resulting chain of edges
```

2.4.1.6. Experimental Results

For a given error threshold Δ_{max} the actually achieved error as well as the achieved reduction in complexity of the polyline may differ from algorithm to algorithm. To avoid erratic interplay of the two measures, the evaluation was carried out as follows: Using a fixed error threshold Δ_{max} the Douglas-Peucker algorithm was applied on the input polyline P in a min-# fashion to obtain an approximation Q_{dp} . The coarsening algorithms were then applied on P in a min- Δ fashion using $|Q_{dp}|$ (which is actually assumed by the coarsening algorithms).¹⁰

The performance of the proposed algorithms was evaluated with respect to two different criteria. Firstly, the quality of the approximation with respect to the input polyline, and secondly, the quality of the approximation with

¹⁰ A given target size k is exactly assumed for the basic coarsening algorithm. For the adaptive coarsening algorithm and the corner simplification algorithm, since they may reject several vertices in a bundle, in some cases the actual number of vertices might be below the target k , however, the impact on the results is supposed to be small.

respect to the original polyline as outlined in Section 2.4.1.1 page 95. The distance between two polylines P and Q was determined as the maximum of the directed Hausdorff distances $d_{\bar{H}}(V(P), Q)$ and $d_{\bar{H}}(V(Q), P)$ with V being the vertex set of a polyline.¹¹

The quality of the approximation with respect to the input polyline was evaluated using three different sets of data: the *MPEG 7* data set (see Section 1.6.2.1), the *UK trademarks* set (see Section 1.6.2.2), and a set of 40 000 computer generated images each of which depicted a randomly distorted version of a simple shape from a list of 40 geometric objects such as, e. g., triangles, quadrilaterals, ellipses, or stylized letters (see Figure 2.18 for an example). The polylines were extracted from the bi-level black-and-white images and redundant vertices were eliminated. For images containing more than one contour, the longest polyline was chosen.

Averaged over 20 different values¹² of the error threshold Δ_{max} , for all three data sets the proposed algorithms performed worse than the Douglas-Peucker algorithm (see Table 2.1 for the results). Having in mind that the Douglas-Peucker algorithm is the preferred simplification method in many application domains, and that the adaptive corner simplification algorithm is specifically wanted to accept additional errors, this is not a surprise. However, looking at the error with respect to the original polylines gives completely different results.

The quality of the approximation with respect to the original polyline was evaluated using the set of computer generated images as described above. The explicitly present polylines of the 40 geometric objects served as original, the polylines extracted from the distorted images served as input for the simplification algorithms. Using exactly the same data and parameters as for the evaluation with respect to the input polylines, compared to the results of the Douglas-Peucker algorithm the averaged errors with respect to the original were 4 % lower for the basic coarsening algorithm, and 6 % lower for the adaptive coarsening plus corner simplification.

The difference between the results of the Douglas-Peucker algorithm and the coarsening algorithms is even more obvious when the error threshold Δ_{max} is chosen based on the—normally unknown—sizes of the features of the original polylines. Let $P^* = (p_1, \dots, p_n)$ be an original polyline and let

¹¹ In the given context this measure is supposed to give a good approximation for the Hausdorff distance $d_H(P, Q)$ and the weak Fréchet distance $d_{wF}(P, Q)$. Malicious configurations for which the values differ can be constructed, however, are not expected to occur frequently—if ever—in the used test data.

¹² The error threshold Δ_{max} was chosen relative to the image size with $\Delta_{max,rel}$ ranging from 0.001 to 0.02.

2. Extraction of Shapes

	with respect to: input P			original P^*
	MPEG 7	UK	generated	generated
basic	1.08	1.05	1.02	0.96
adaptive + corner	1.79	1.45	1.50	0.94

Table 2.1. Performance of simplification algorithms: average approximation errors of the basic coarsening algorithm and of the adaptive coarsening plus corner simplification relative to results of the Douglas-Peucker algorithm.

$s(P^*) := \min_{1 < i < n} \{ \text{dist}(p_i, \overline{p_{i-1}p_{i+1}}) \}$ denote the minimum distance of any vertex p_i to the segment spanned by its neighbors in P^* . Since the definition of $s(P^*)$ corresponds to the local error criterion δ^{ms} used by the algorithms under consideration, it gives an upper bound on the error threshold for which simplifying P^* leaves it unchanged and therefore does not yield errors at all.

Choosing the error threshold Δ_{max} relative to $s(P^*)$ shows that while preserving the original features, the complexity of the approximating polyline can be reduced more efficiently using the adaptive coarsening plus corner simplification than using the Douglas-Peucker algorithm. Figure 2.21 shows the average results for the set of 40 000 generated polylines.

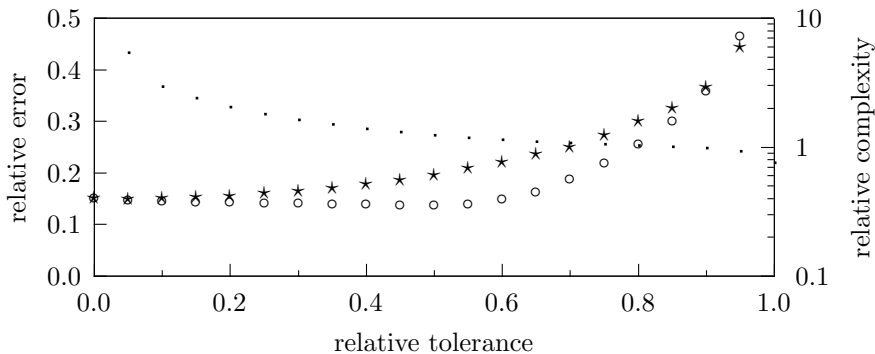


Figure 2.21. Performance of simplification algorithms: errors relative to feature size for Douglas-Peucker algorithm (★), and for adaptive coarsening plus corner simplification (○), plus complexity of approximating polylines relative to original (·).

The actually achieved relative error of the Douglas-Peucker algorithm starts to increase early and exceeds the value of 0.16 at a relative error tolerance of 0.25 where the relative complexity of the approximating polylines is 1.76. The actually achieved relative error of the adaptive coarsening plus corner simplification on the other hand stays small until the (corresponding) relative error tolerance reaches 0.50 where the relative complexity of the approximating polylines is 1.21. That means, that while ensuring the same small approximation error, the adaptive coarsening plus corner simplification produces polylines that on the average have 0.69 times the number of vertices compared to polylines produced by the Douglas-Peucker algorithm. Only for values of the relative error tolerance that should be avoided because of the risk of destroying original features with either algorithm (near 1.0), the actually achieved relative errors of the adaptive coarsening plus corner simplification exceed the values of the Douglas-Peucker algorithm.

2.4.2. Deletion of Irrelevant Data

Erroneous pixels in the raster graphics as well as variations of color in conjunction with deficiencies of the vectorization process may lead to the detection of shapes that, however, are not perceived at all or that are not perceived as being relevant. In order to reduce the complexity of the image descriptions these shapes should be removed.

2.4.2.1. Isolated Small Shapes

A sufficiently small shape—if not contributing to texture or to a line shape (see Section 2.3)—is normally considered to result from noise in the image. However, whether a shape is *sufficiently* small does not only depend on the size, but also on the arrangement of shapes. A shape that is perceived as being noise when isolated, may be perceived as being a feature of the image when close to other shapes (see Figure 2.22 for an example). However, if the union of neighboring small shapes itself is not sufficiently large, the shapes may still be perceived as noise.



Figure 2.22. Noise vs. feature: three dots, two of them perceived as being eyes in a face, one rather perceived as being noise.

2. Extraction of Shapes

In the present work this observation is factored in by removing noise shapes depending on a threshold θ_s on the minimum size of shapes, and a threshold θ_c on the minimum size of groups of shapes: First, shapes with diameter smaller than θ_s are removed. Then, based on the minimum distance between shapes and a threshold on this distance, the shapes are grouped according to a single linkage clustering.¹³ Each shape belonging to a group with diameter smaller than θ_c is also removed.

2.4.2.2. Subordinate Shapes

A sufficiently small or sufficiently thin shape completely lying in a thin neighborhood of the border of a larger shape is normally perceived as being a feature of the borderline (increased line thickness, adumbrated shadow effect etc.), rather than as a relevant shape. Moreover, if a shape is depicted by its outline, but due to the connectedness of the pixels has been detected as a region with a hole, the resulting couple of two almost equal shapes may be replaced by a single shape. These facts may be exploited to further reduce the complexity of an image representation: Let θ_n be a threshold on the maximum distance. Every shape S such that its boundary is in the θ_n -neighborhood of the boundary of a larger region shape R , meaning $d_{\bar{H}}(\partial(S), \partial(R)) < \theta_n$, is removed.

¹³ Let $G = (V, E)$ be the graph that has the set of (region) shapes under consideration as vertex set V , and an edge for each two shapes with minimum distance smaller than the threshold, or formally $E = \{\{S_1, S_2\} \mid \exists p_1 \in \partial(S_1), p_2 \in \partial(S_2) \text{ such that } \|p_1 - p_2\| < \theta_d(S_1, S_2)\}$ for $\theta_d(S_1, S_2)$ being the threshold on the distance for shapes S_1 and S_2 . The single linkage clusters correspond to the connected components in G .

2.4.2.3. Experimental Results

Figure 2.23 shows the simplified resulting shapes for the exemplary image from Figure 2.3.

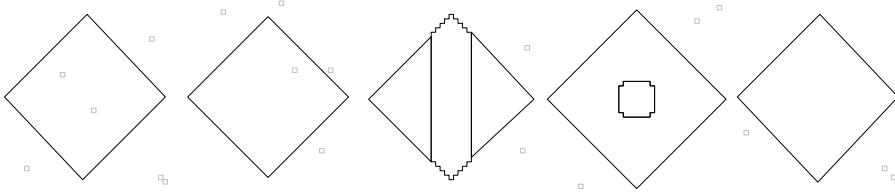


Figure 2.23. Irrelevant data—artificial image:
resulting shapes (black) after small and subordinate
shapes (gray) have been removed.

Applying the algorithms to the *UK trademarks* set, isolated small shapes have been removed in 7 025 (65.4%), and subordinate shapes have been removed in 3 515 (32.7%) of the 10 745 image files. Averaged over all files, 62.4 isolated small shapes and 3.0 subordinate shapes per file—summing up to 16.0% of the total number of shapes originally detected in the images—have been removed. That means, that the complexity of the image descriptions is remarkably reduced.

The *Aktor set* (see Section 1.6.2.3) also contains color images for which boundaries between shapes are not as clear cut as for bi-level black-and-white images. The number of subordinate shapes detected in images of this set, therefore is considerably larger: subordinate shapes have been removed in 16 266 (77.9%) of the 20 894 image files.

2.5. Grouping

Since the shapes perceived in an image may be composed of smaller parts (see Section 1.1.3 and Figure 1.3 on page 1.3), in many applications it is desirable to group parts of the detected boundaries together such that they form the shapes essentially depicted. Of course it might be, that parts may be grouped in several ways. The boundaries in Figure 2.24 (a), e. g., can be seen as forming four separate squares or forming a single cross-like shape with a hole inside.

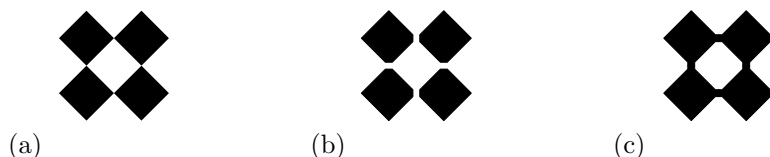


Figure 2.24. Indeterminate Grouping:
 (a) squares touching, (b) squares separated, (c) squares merged.

Theoretically the number of possible groupings may be exponential in the number of parts, however, there are general theories describing tendencies of how humans group boundary parts together (see Section 1.2.2). In [200] an automated approach for grouping lines in sketches and drawings was presented. It uses a weighted directed graph where the vertices represent the boundary parts and the edges represent possible linkages of two parts. Based on the local geometry of the parts, the edges get weights according to predefined scores reflecting the perceptual preferences of grouping. The same approach—with slightly adapted preference scores—has also been used for figurative images in [113].

If the task is to determine whether a specific shape is depicted in an image (as, e. g., in Section 4.4), such an approach may be quite beneficial. However, if the task is to find a good representation for similarity estimation, which representation is the best heavily depends on the other images. Using all possible representations is computationally infeasible, but if the similarity measure is defined on single shapes, choosing one representation—or a limited number of representations—a priori, entails the risk of rejecting a representation that would have been needed for proper similarity estimation. Figure 2.24 shows an example: deciding for representing image (a) by four squares is entirely suitable for comparing it to the four square-like shapes in image (b), but absolutely not for comparing it to the cross-like shape in image (c) and vice versa. However, this dilemma may be resolved by using similarity measures that are robust with respect to different representations (see Section 3.3).

Estimation of Shape-Similarity

This chapter deals with the problem of estimating perceived similarity of figurative images based on the sets of depicted shapes. The shapes considered here are (polygonal) line shapes and (polygonal) region shapes in the plane as defined in Section [1.1.2](#).

3.1. Basics

There are various ways in which two figurative images may be perceived to be similar: In the most obvious case the second image is almost a copy of the first image—this copy may be translated, scaled, or rotated— but it may also only contain a part that is almost a (translated, scaled, or rotated) copy of (a part of) the first image. These aspects of similarity are closely related to classical object recognition. However, in similarity estimation transformations other than rigid motions, scalings and projections also have to be considered. Moreover, there might also be further (shape-related) features that lead to the perception of similarity, e. g., symmetry, connectedness, etc. [[222](#)].

In order to estimate the similarity of figurative images based on the sets of depicted shapes, a suitable representation of these sets of shapes has to be found. On the one hand such a representation should capture the essence of the perception, on the other hand it should be of low complexity. The question how the essence of a shape can appropriately be described has not yet been answered satisfactorily. The fact that a shape may comprehensively be characterized by its boundary, and the observation that perceived information about a shape is concentrated along its contour (see Section [1.2.2](#)), may lead

3. Estimation of Shape-Similarity

to the assumption that the interior of a shape can totally be ignored. However, in [168] it is claimed that no successful theory of shape description can ignore either the boundary or the interior—Figure 3.1 shows an example underpinning this claim: Although the boundaries of the two shapes equal each other to a large extent, the shapes are not perceived to be very similar because the different spatial arrangement of the boundary parts induce different interiors.

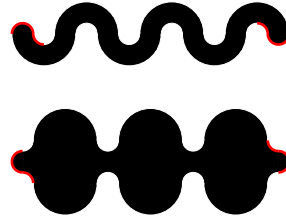


Figure 3.1. Interior vs. contour:

The depicted shapes have almost the same contours—which differ only in the parts marked red—but due to their interior they are perceived differently.

There are primarily two ways to reduce complexity which lead to different types of representations, namely selective representations and accumulative representations. Selective representations use a limited number of features of equal type selected from a larger set of these features, e. g., a limited number of points from the shapes boundaries. Accumulative representations, on the other hand, use information gathered by summarizing properties over a shape or even the whole set of shapes, e. g., the average distance of shape points from the center.

Selective attention (see Section 1.3.3.2) and the need to detect also partial similarities pose problems for both of these approaches. By selecting a limited number of features, too many features originating from the part that contributes to the perception of similarity might be dismissed because of a large total number of features. By accumulating over the whole set of shapes¹ the resulting representation might be totally different from the one for the part that contributes to the perception of similarity.

Further reasons why defining representations suitable for similarity estimation of shapes or of sets of shapes is problematic, come from the fact that shapes may be depicted in different ways. For instance, Figure 2.1 on page 68 shows a sharply bent strip that is perceived similar to (the outline of) a square. Figure 2.24 on page 110 shows sets of shapes that are topologically different while they are perceived similar. Moreover, even single shapes that almost equal each other geometrically may have very different internal structures (see Figure 3.2).

¹ The same effect may also occur for single shapes, of course.

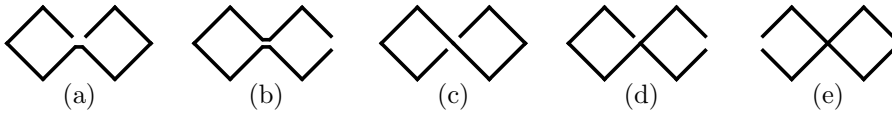


Figure 3.2. Difficulties in defining suitable shape representations:
Shapes differing in various ways, but perceived as being very similar.

Perceived similarity can be assumed to be invariant with respect to asynchronous transformation under translations and scalings. Representations used in the context of similarity estimation, therefore, are often wanted also to be invariant under translations and scalings. This is often achieved by normalizing position and size of the input before computing the representation. This normalization can, e.g., be done with respect to the smallest axis aligned rectangle containing the input as in [132], or with respect to the smallest circle containing the input as suggested in [114]. However, if the estimation of the reference frame may be influenced by other shapes or even by noise, this heavily limits the applicability of such a normalized representation.

In order to safely preclude undesirable effects of all these issues, the approach for estimating the similarity between sets of shapes presented in the following, uses the boundary polygons of the region shapes plus polygonal lines just as they are given. It works in two phases: Firstly, promising transformations for aligning the two shape sets are determined. Secondly, the similarity of the aligned sets of shapes is estimated using a similarity measure that, therefore, does not need to be invariant with respect to asynchronous transformation.

The problems of determining an optimal transformation, or finding a constant factor approximation, have extensively been studied for commonly used measures of (dis-)similarity. However, most of the results are restricted to a very limited class of transformations, namely translations, or they are restricted to a very limited class of possible inputs, e.g., convex shapes. Here, while making no guarantees on optimality, the input may consist of arbitrary sets of polygonal shapes, and the classes of allowable transformations include—among others—translations, similarity transformations, and even generic affine transformations.

3.2. Mapping

This section addresses the following problem: Given two sets \mathcal{P}_1 and \mathcal{P}_2 of polylines in the plane, and a class \mathcal{T} of allowable transformations from the \mathbb{R}^2 to the \mathbb{R}^2 , find a transformation $t \in \mathcal{T}$ mapping \mathcal{P}_1 onto \mathcal{P}_2 . The proposed approach computes transformations that comply with an intuitive notion of *matching*, that is, they map parts of one polyline set close to similar² parts of the other polyline set.

3.2.1. Related Work

The mapping problem is often stated as follows: Given a model M (in the given context a set of shapes) which is described by a set F_M of m features,³ an image D from which a set F_D of n features has been extracted, and a class \mathcal{T} of allowable transformations, detect an occurrence of the model in the image by finding a transformation $t \in \mathcal{T}$ that maps as many features of F_M close to (corresponding) features of F_D as possible. There are two types of errors that have to be considered in this context. Firstly, positional errors of the correctly detected features. Secondly, so called outliers (as opposed to inliers) which are either features present in the model that have not been detected in the image, or features that spuriously have been detected in the image although they are not present in the model. Various methods to solve the mapping problem have been proposed, which differ in the ways candidate transformations are generated and how they are rated:

Alignment Methods and the Random Sample Consensus Let $k < m$ be the minimum number of feature pairs from $F_M \times F_D$ uniquely defining a transformation from \mathcal{T} that exactly maps the pairs' first elements to the corresponding second elements. The idea behind alignment methods as described in [116, 117] is to use a k -tuple of features from F_M and a k -tuple of features from F_D to form a sample set $S \subset F_M \times F_D$ that defines a transformation t_S , to apply the transformation to the model features and to rate the transformation based on some measure of similarity on the set of transformed model features and the set of image features. In the original version, every possible k -set of feature pairs has been considered independently. There are $\binom{m \cdot n}{k} = \Theta((m \cdot n)^k)$ such sets, but for any proper occurrence of the model in the image $\binom{m}{k} = \Theta(m^k)$ sample sets result in combinatorially equivalent transformations which are then unnecessarily re-rated.

² Here, conflicting with the demand formulated in Section 1.3.1, no specific measure of similarity is referred to.

³ Of course also the features detected in an image may serve as a model.

In order to reduce the number of re-ratings, it has been proposed to randomize the selection of the sample sets, for instance by applying the so called *random sample consensus* (RANSAC) as introduced in [84]. The random sample consensus is a general paradigm for determining the parameters of an arbitrary model (not necessarily shapes) from a given set P of data points. Again, let k be the minimum number of data points needed to determine the free parameters of a given model. The random sample consensus works as follows: Repeatedly a random set S of k sample points from P is selected, based on that set an instantiation M_S of the model is computed such that it exactly fits S , and the so called *consensus set* C_S , the set of points in P that are within some error tolerance of M_S , is determined.⁴ Using the consensus set of largest cardinality the instantiation of the model is recomputed, or if the largest cardinality is too small at all it is assumed that no instantiation is present in the data.

In [172] the random sample consensus has been applied to the mapping problem in the following way: Repeatedly a random tuple S_M of k features from F_M is selected, for every tuple S_D of k features from F_D the corresponding transformation $t_{S_M S_D}$ is computed and applied to the model features in order to compute the cardinality of the consensus set. Given the ratio α of inliers in F_M , the failure probability is below a threshold θ if the number of rounds is greater or equal $\lceil \frac{\log \theta}{\log(1-\alpha^k)} \rceil$. This means that for a constant bound on the failure probability the number of transformations considered is in $O(k! \cdot n^k)$. For an unknown ratio of inliers an adaptive algorithm is also given. In [122] the overall running time on realistic inputs is reduced even more by choosing a random set of model features for estimating the consensus set. All these analyzes, however, only take the existence of outliers into account but do not consider positional errors.

Geometric Hashing Given a class \mathcal{T} of allowable transformations, the idea behind geometric hashing as introduced in [142, 141] is to use a representation of the set of model features F_M that is invariant under the class \mathcal{T} . This is achieved by expressing each feature relative to a coordinate frame defined by a tuple of model features. For example, in the case of translations a single point p_0 may serve as reference and every other point p is then uniquely determined by the relative coordinates $(p - p_0)$. In the case of affine transformations three (non-collinear) points p_0, p_1 , and p_2 may serve as reference. Every other point p is then uniquely determined by two coordinates λ_1, λ_2 such that $p = p_0 + \lambda_1 \cdot (p_1 - p_0) + \lambda_2 \cdot (p_2 - p_0)$. A very similar idea was also suggested in the context of comparison of biometric landmarks [33].

⁴ The authors also propose a variation of their paradigm, namely that the sample set S might be enlarged by adding all new data points of the consensus set C_S to S and that the model might then be recomputed based on this enlarged set. A similar idea will be pursued in the present work.

3. Estimation of Shape-Similarity

Let k be the number of points needed to define such a reference frame invariant under the class \mathcal{T} . Geometric hashing works in two stages: preprocessing of the model, and voting. In the preprocessing stage every k -tuple R_M of features from F_M is used as reference, the coordinates with respect to R_M of each remaining feature from $F_M \setminus R_M$ are computed and (after quantization) are used as key for storing the reference R_M in a hash table. In the voting stage every k -tuple R_D of features from F_D is used as reference, the coordinates with respect to R_D of each remaining feature from $F_D \setminus R_D$ are computed and (after quantization) are used as key for looking up the corresponding reference tuples of the model in the hash table. If a reference tuple R_M of the model gets sufficiently many votes from a reference tuple R_D of the image, the transformation exactly mapping R_M to R_D is assumed to indicate an occurrence of the model in the image.

The preprocessing needs $\Omega(m^{k+1})$ time and space, but the voting can be done in $O(n^{k+1})$ time, independent from the complexity of the model. Moreover, for a given threshold on the probability of failure, the running time of geometric hashing also can be decreased by randomization [122]. However, just as for the alignment methods, only the existence of outliers is taken into account but the effects of positional errors are neglected.

The Generalized Hough Transform and Pose Clustering In its original version, the Hough transform as described in [115] was used to recognize straight lines in images, but the idea has also been generalized to the recognition of other shapes (see [68], [163], and [20]).

The (generalized) Hough transform determines the free parameters of a given model (for example slope and distance from origin for straight lines, coordinates of the center and radius for circles, or the parameters describing the pose of an explicitly given shape) by gathering evidences in the space of free parameters. A bounded region of the parameter space is usually discretized by some grid, each cell acting as an accumulator for evidences. For each sample of image features (in the original version just single points) every cell corresponding to a feasible set of parameters receives a vote. Cells with a sufficiently large number of votes are supposed to correspond to occurrences of the model in the image.

The Hough transform in its original version uses samples of image features (points) such that for each sample the corresponding parameters of the model are under-determined. Hence, for each sample the space of feasible parameters has dimension greater than zero and a whole set of accumulators receives votes. Due to the trade-off between effort in image space and parameter space as discussed in [20], most generalizations of the Hough transform, however, use samples such that for each sample the feasible parameters are uniquely defined and form a single point in the parameter space.

The term *pose clustering* is used for generalizations of the Hough transform that assume the model to be given as a set of features plus a class of allowable transformations, and that use samples such that each sample uniquely defines a transformation (see, e. g., [213] and [173]). These approaches are closely related to geometric hashing, however, the way the votes are accumulated is different.

The idea of using samples from a set of features, to compute the corresponding transformations, and to cluster these transformations has also been successfully applied for the problem of symmetry detection [165].

Probabilistic Shape Matching The features considered in probabilistic shape matching as described in [201] are single points of the shapes. Let k be the minimum number of point pairs from $F_M \times F_D$ uniquely defining a transformation from \mathcal{T} that exactly maps the pairs' first elements to the corresponding second elements. Probabilistic shape matching applies the same mechanisms as pose clustering: repeatedly a k -tuple of feature points from F_M and a k -tuple of feature points from F_D is chosen to form a sample set $S \subset F_M \times F_D$, the corresponding transformation t_S constitutes a vote, and clusters of votes (small regions in transformation space with sufficiently many votes) are assumed to indicate an occurrence of the model in the image. However, probabilistic shape matching differs in the generation of the samples: Unlike the other methods it does not assume finite sets of features, but uses randomly chosen points of the shapes—in [201] of polygonal curves in the plane, in [8] of arbitrary two-dimensional regions in the plane.

3.2.2. Idea of the Proposed Approach

All the methods listed above have in common that they use samples of minimum cardinality to determine a hypothesis (transformation) which gets either evaluated directly, or is gathered up to form a cluster. This minimality, however, implies that the data used to determine a hypothesis does not contain any redundancy and that the hypotheses are therefore maximally prone to positional errors. In the context of object recognition the positional errors are usually small, however, in the context of similarity estimation also distortions have to be considered.

One possible way (suggested, e. g., in [84]) to reduce the impact of individual positional errors is to use larger samples that contain redundancies.⁵ The courses of the polygonal curves may be very helpful in identifying larger consistent sets of feature pairs. On the other hand, different representations of essentially the same shape (see Figures 1.11 on page 50 and 2.24 on page 110 for examples) may even thwart the identification.

⁵ For the problem of recognizing 3d objects in 2d images using point samples, a detailed analysis of the impact of positional errors, and of the improvements achieved by enlarging the samples is given in [9].

3. Estimation of Shape-Similarity

In order to achieve both—robustness against combinatorial differences and robustness against positional errors—the approach presented in the following combines the voting of probabilistic shape matching with enlarging the samples: Repeatedly an initial sample of minimum cardinality is drawn randomly from the shapes, the corresponding transformation is computed and the sample is iteratively enlarged while the corresponding transformation is updated. The enlarging of the sample is stopped when the gathered data becomes inconsistent. In this way each initial sample constitutes the starting point for a sequence of sample-transformation pairs, the best of which is weighted according to the size of the sample and the quality of the mapping. The weighted votes are clustered according to a distance measure defined on transformation space, and clusters with large total weight are assumed to indicate the existence of correlating parts in the two shape sets. Using enlarged samples does not only facilitate robustness against positional errors, but it also reduces the number of necessary votes drastically compared to probabilistic shape matching in its basic form. The structure of the proposed approach is shown in Algorithm 3.1. A detailed description is given in the following sections and a set of suitable values for the parameters described there is given in the Appendix (Section B.2).

Algorithm 3.1:

```
 $T \leftarrow \emptyset$ 
for number of samples do
   $S_0 \leftarrow$  (random vertex of first set, random vertex of second set)
   $t_{best} \leftarrow t_{S_0}$ 
  repeat
     $S_i \leftarrow$  grow( $S_{i-1}$ )
    if  $t_{S_i}$  better than  $t_{best}$  then
       $t_{best} \leftarrow t_{S_i}$ 
    end
  until  $S_i$  inconsistent
  compute weight of  $t_{best}$ 
   $T \leftarrow T \cup \{t_{best}\}$ 
end
compute clusters of  $T$ 
```

3.2.3. Sampling

Idea Conspicuous features of shapes arise from (boundary) regions of high curvature [17]. Regarding polylines, these regions are the vertices. However, not every vertex—even though its turning angle may be large—needs to constitute a feature recognizable by a human observer. Therefore, the samples will primarily be drawn from the sets of polyline vertices, but also points in the interior of edges will be considered if they correspond to vertices of the other polyline. Given a polyline $P \in \mathcal{P}_1$ and a polyline $Q \in \mathcal{P}_2$, starting from a randomly chosen vertex of P and a randomly chosen vertex of Q , the polylines are explored in both directions and pairs of corresponding vertices (or vertex surrogates) are added to the sample. When the sample starts to become inconsistent the exploration is stopped and the best sample set found so far is used.

Initial Samples For a set \mathcal{P} of polylines, let $V(\mathcal{P})$ be the union of the sets of vertices of the polylines in \mathcal{P} . An initial sample S_0 consists of a randomly chosen vertex p_i from a polyline of \mathcal{P}_1 and a randomly chosen vertex q_j from a polyline of \mathcal{P}_2 . On the one hand, choosing vertices uniformly at random would immoderately favor small complex structures. On the other hand, choosing vertices according to the length of the adjacent edges (which corresponds to uniformly choosing any point of the polyline like in probabilistic shape matching, and then taking the next vertex) would immoderately favor vertices of large, simple structures like, e.g., rectangular frames (see Section 1.2.3 page 28 for a discussion). In order to keep these two effects in balance, the two strategies are combined: For a vertex $v \in V(\mathcal{P})$ let $\omega(v)$ be the sum of the length of the two edges incident to v and let $\omega_{sum} = \sum \omega(v)$ be the sum of these values. Furthermore let n be the number of vertices of \mathcal{P} . Each vertex v is chosen with probability $0.5 \cdot 1/n + 0.5 \cdot \omega(v)/\omega_{sum}$. In this way, informally speaking, half of the vertices are chosen according to the complexity and half of the vertices are chosen according to the size.

If the class of allowable transformations does permit scalings, also a prescaling factor c is randomly chosen, based on which further correspondences for the augmentation of the initial sample will be determined. Since similar shapes presumably contain similar configurations of features, a factor c_s is determined based on the initial sample (p_i, q_j) and two additional random vertices $p_{i'} \in V(\mathcal{P}_1)$ and $q_{j'} \in V(\mathcal{P}_2)$ as $c_s := \|q_j - q_{j'}\| / \|p_i - p_{i'}\|$ such that $c^{min} \leq c_s \leq c^{max}$ for given thresholds on the minimum and maximum scaling of a reasonable transformation. Since the configurations of features may slightly differ, also a purely random factor c_r is chosen such that $\log(c_r)$ is normally distributed with mean zero and small variance. The overall prescaling factor is then defined as $c := c_s \cdot c_r$.

3. Estimation of Shape-Similarity

Augmenting a Sample Let $P = (p_1, \dots, p_m)$ and $Q = (q_1, \dots, q_n)$ be two polylines. If the part $(p_{a_1}, \dots, p_{a_2}) \subset P$ corresponds to the part $(q_{a_3}, \dots, q_{a_4}) \subset Q$, the predecing vertices p_{a_1-1}, q_{a_3-1} and the succeeding vertices p_{a_2+1}, q_{a_4+1} would surely be candidates for finding additional correspondences. However, a vertex need not necessarily have a corresponding vertex on the other polyline, but may also correspond to a point in the interior of a long edge (see Figure 3.3).

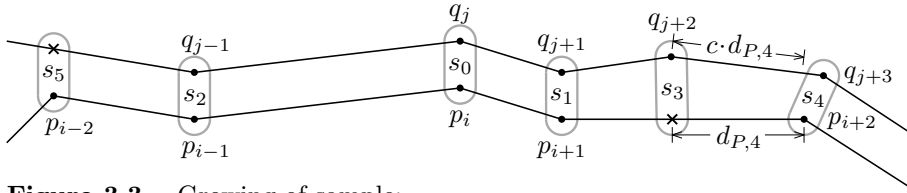


Figure 3.3. Growing of sample: polyline vertices (\bullet), and vertex surrogates (\times) for growing an initial sample $\{s_0\} = \{(p_i, q_j)\}$ to $\{s_0, s_1, \dots, s_5\}$.

The initial sample S_0 contains only the pair s_0 consisting of a vertex $p_i \in P$ and a vertex $q_j \in Q$ randomly chosen as described above. Furthermore, a forward direction for exploring P and a forward direction for exploring Q (without loss of generality the forward direction will be assumed to correspond to ascending indices in the following) are randomly chosen. Let $s_f = (s_{f,1}, s_{f,2})$ be the last sample added to S originating from a forward exploration (initially $s_f = s_0$) and let d_f accumulate the distances on Q explored in forward direction (initially $d_f = 0$). Accordingly, let $s_b = (s_{b,1}, s_{b,2})$ be the last sample added to S originating from a backward exploration and let d_b accumulate the distances on Q explored in backward direction. Lengths concerning P and lengths concerning Q are compared based on the assumption that the transformations mapping P to Q isotropically scale all lengths with factor c .⁶

In the case of forward exploration, let p be the vertex on P succeeding to $s_{f,1}$ and let d_P be the scaled distance $c \cdot \|p - s_{f,1}\|$. Accordingly, let q be the vertex on Q succeeding to $s_{f,2}$ and let $d_Q := \|q - s_{f,2}\|$. If $|d_P - d_Q| \leq \theta^a \cdot \min(d_P, d_Q)$ for a threshold θ^a (meaning that the relative difference between the two distances is sufficiently small) the vertices p and q are assumed to be corresponding and form a new sample pair. The new pair replaces s_f , is added to S , and d_f is incremented by d_Q . If the relative difference between the two distances is larger than the threshold, then a vertex surrogate is introduced (see

⁶ For general affine transformations this need not be the case, but see Section 3.2.4 page 128 for a discussion.

Figure 3.3 for an example): In case $d_P < d_Q$ the vertex p has no corresponding vertex on Q and therefore a point on the edge from $s_{f,2}$ to q with distance d_P from $s_{f,2}$ is determined as $q' := (1 - d_P/d_Q) \cdot s_{f,2} + (d_P/d_Q) \cdot q$. The vertex p and the vertex surrogate q' form a new sample pair which replaces s_f and is added to S . The accumulated distance d_f is incremented by d_P . Accordingly, in case $d_Q < d_P$ a vertex surrogate on P is determined as $p' := (1 - d_Q/d_P) \cdot s_{f,1} + (d_Q/d_P) \cdot p$ and the pair (p', q) replaces s_f and is added to S . The accumulated distance d_f is incremented by d_Q then.

Backward exploration works accordingly. Exploration is always performed in the direction for which the accumulated distance is smaller, unless the end of a polyline is reached in this direction. The augmentation of the sample is stopped when forward and backward exploration both reached endpoints of a polyline (or in case of closed polylines, meet in a common vertex) or when the sample starts becoming inconsistent.

Checking Consistency The consistency of a sample S is rated based on a constant number of pairs of corresponding points, namely on s_0 and, since inconsistencies may only be introduced by newly added pairs, on s_f and s_b . Let t_S be the transformation mapping P to Q that has been computed based on the sample S (as will be described below), then the error measure used to check the consistency is $\bar{\delta}'(S) := (\|s_{0,2} - t_S(s_{0,1})\| + \|s_{f,2} - t_S(s_{f,1})\| + \|s_{b,2} - t_S(s_{b,1})\|)/3$.

Whether a sample is consistent, however, cannot be decided on the value of this error alone: For samples representing large explored parts of polylines higher values may be tolerated than for a sample representing a short single edge. One important factor influencing the size of the maximum tolerated error is the spatial spread of the sample measured by the sum d_{box} of the side lengths of the smallest axis aligned rectangle containing the sampled points from Q . If the sampled part of a polyline grows, but stays in the same bounding box, however, the sample also gains significance and therefore also the length $d_f + d_b$ of the explored part is considered.

Given a threshold θ^e , the relative error that is still tolerated is computed as $\delta^{tol}(S) := \theta^e \cdot (d_f(S) + d_b(S) + d_{box}(S))$. In this way, the augmentation of the sample induces a sequence of tolerable errors and a sequence of actually occurring errors. The sample is considered to be inconsistent and the augmentation of the sample is stopped if $\bar{\delta}'(S) > \delta^{tol}(S)$. However, at this point unsuitable data has already been incorporated into the sample—unless it would not be inconsistent. The sample that is chosen, therefore, is the one corresponding to the largest difference between tolerable error and actual error (see Figure 3.4 for an illustration).

3. Estimation of Shape-Similarity

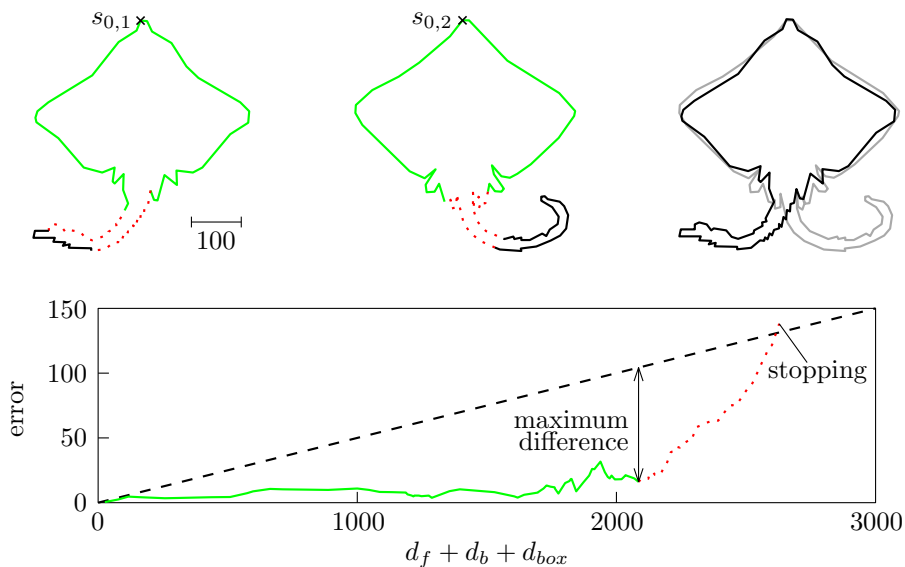


Figure 3.4. Checking consistency:
 (top) parts explored during the augmentation of a sample for determining a translation, plus the shapes superimposed using the resulting translation; parts contributing to the final sample in green, parts explored but not contributing in dotted red;
 (bottom) corresponding plot of actual error $\bar{\delta}'$ in green/dotted red, plus tolerable error δ^{tol} in dashed black.

3.2.4. Computing Transformations

Let \mathcal{T} be a class of transformations from the \mathbb{R}^2 to the \mathbb{R}^2 and let k be the minimum number of point pairs from $\mathbb{R}^2 \times \mathbb{R}^2$ needed to uniquely determine a transformation from \mathcal{T} that maps the pairs' first points exactly to the corresponding second points. For a set $\{(s_{1,1}, s_{1,2}), \dots, (s_{n,1}, s_{n,2})\}$ of $n > k$ pairs, in general there is no such transformation exactly mapping every $s_{i,1}$ to the corresponding $s_{i,2}$ or in other words, there is no transformation $t \in \mathcal{T}$ such that $\|t(s_{1,1}) - s_{1,2}\| = \dots = \|t(s_{n,1}) - s_{n,2}\| = 0$.

The basic approach applied in the following is to minimize the sum of the squared errors. As introduced in [145] and [93], this method can be used to determine the parameters of a model from measured values that contain random errors. In the context of the present work, the distortions are mostly *not* caused by random errors, however, the results achieved by applying the

method of least squares are convincing. Compared to minimizing the maximum error, the method of least squares has the advantage that the point pairs with small errors are not ignored, but do also contribute to the result. In fact this is favorable when a transformation is looked for, that maps as much of the edges of one set of polylines as close as possible to (corresponding) edges of the other set of polylines.

Treating all sample pairs as equally important could lead to counterintuitive results, because the perceived importance of a part of a polyline is first of all not determined by the number of vertices—which also may heavily depend on the actual representation—but on the expanse. The sample pairs are therefore weighted according to the lengths of the incident edges (that have been explored so far). Let s_{a_1} , s_{a_2} , and s_{a_3} be three sample pairs such that $a_1 < a_2 < a_3$ and the points $s_{a_1,2}$, $s_{a_2,2}$, and $s_{a_3,2}$ appear consecutively on polyline Q . Before pair s_{a_3} is added to the sample set, the pair s_{a_2} is weighted with $\|s_{a_1,2} - s_{a_2,2}\|/2$. After pair s_{a_3} has been added to the sample set, the pair s_{a_2} is weighted with $\|(s_{a_1,2} - s_{a_2,2})\|/2 + \|(s_{a_2,2} - s_{a_3,2})\|/2$. Let ω_i be the weight for sample pair s_i in the sample set S and let $n = |S|$. The transformation computed for S is $t_S = \arg \min_{t \in \mathcal{T}} \sum_{i=1}^n (\omega_i \cdot \|t(s_{i,1}) - s_{i,2}\|^2)$.

For all the classes of transformations considered in the following it is possible to organize the calculation of the optimal transformations for a sequence (S_1, \dots, S_m) of sample sets with $|S_{i+1}| = |S_i| + 1$ and $S_i \subset S_j$ for $i < j$, such that having computed the optimal transformation t_{S_i} , the optimal transformation $t_{S_{i+1}}$ can be computed in constant time.⁷

All the classes of transformations considered here are (not necessarily proper) subclasses of affine transformations. That means that such a transformation t can be characterized by a 2×2 -matrix M and a translation vector v , such that for a point p

$$t(p) = (M \cdot p) + v = \begin{pmatrix} M[1][1] \cdot p[x] + M[1][2] \cdot p[y] + v[x] \\ M[2][1] \cdot p[x] + M[2][2] \cdot p[y] + v[y] \end{pmatrix}.$$

Given a sample set $S_n = \{(s_{1,1}, s_{1,2}), \dots, (s_{n,1}, s_{n,2})\}$, in the following, $x_{i,1} := s_{i,1}[x]$ denotes the x-coordinate of the i -th sample pair's first point, $y_{i,1}$ the y-coordinate and analogously $x_{i,2}$ and $y_{i,2}$ denote the coordinates of the corresponding second point.

⁷ Please note that also the change of weight for the predecesing sample pair has to be considered, but this can also be dealt with in constant time.

3. Estimation of Shape-Similarity

Translations For translations, M is the identity matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and v is an arbitrary vector. Since there are only 2 degrees of freedom, k equals 1. The translation vector of the translation that minimizes the sum of the weighted squared distances for the sample set $S_n = \{(s_{1,1}, s_{1,2}), \dots, (s_{n,1}, s_{n,2})\}$ can easily be computed as

$$v = \frac{1}{\omega(S)} \cdot \sum_{i=1}^n \omega_i (s_{i,2} - s_{i,1})$$

with $\omega(S)$ being the sum of all weights.

Homotheties A homothety is a combination of a uniform scaling and a translation. M is the matrix $\begin{pmatrix} c & 0 \\ 0 & c \end{pmatrix}$ with c being the scaling factor, and v is an arbitrary vector. Since there are 3 degrees of freedom, at least 2 sample pairs are needed to define a homothety, however, in general there is no homothety exactly mapping the pairs' first points to the corresponding second points. The homothety that minimizes the sum of the weighted squared distances for the sample set $S_n = \{(s_{1,1}, s_{1,2}), \dots, (s_{n,1}, s_{n,2})\}$ can easily be computed by solving the following system of linear equations:

$$\sum_{i=1}^n \omega_i \begin{pmatrix} x_{i,1}^2 + y_{i,1}^2 & x_{i,1} & y_{i,1} \\ x_{i,1} & 1 & 0 \\ y_{i,1} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c \\ v[x] \\ v[y] \end{pmatrix} = \sum_{i=1}^n \omega_i \begin{pmatrix} x_{i,1}x_{i,2} + y_{i,1}y_{i,2} \\ x_{i,2} \\ y_{i,2} \end{pmatrix}.$$

Rigid Motions A rigid motion is a combination of a rotation and a translation. M is the matrix $\begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix}$ with φ being the angle of rotation around the origin, and v is an arbitrary vector. Since there are 3 degrees of freedom, at least 2 sample pairs are needed to define a rigid motion, however, in general there is no rigid motion exactly mapping the pairs' first points to the corresponding second points.

The rotation matrix M' and the translation vector v' of the rigid motion that minimizes the sum of the *unweighted* squared distances for the sample set $S_n = \{(s_{1,1}, s_{1,2}), \dots, (s_{n,1}, s_{n,2})\}$ can be computed as described in [43]: Let $\bar{s}_1 = \frac{1}{n} \sum_{i=1}^n s_{i,1}$ and $\bar{s}_2 = \frac{1}{n} \sum_{i=1}^n s_{i,2}$ denote the centers of mass of the sets

of sample points and let $\hat{s}_{i,1} = s_{i,1} - \bar{s}_1$ and $\hat{s}_{i,2} = s_{i,2} - \bar{s}_2$ be the coordinates relative to these centers. Let furthermore $a = \sum_{i=1}^n (\hat{x}_{i,1}\hat{x}_{i,2} + \hat{y}_{i,1}\hat{y}_{i,2})$ and $b = \sum_{i=1}^n (\hat{x}_{i,1}\hat{y}_{i,2} - \hat{y}_{i,1}\hat{x}_{i,2})$, then

$$M' = \frac{1}{\sqrt{a^2 + b^2}} \cdot \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \quad \text{and} \quad v' = \bar{s}_2 - M' \cdot \bar{s}_1.$$

The rotation matrix M and the translation vector v of the rigid motion that minimizes the sum of the *weighted* squared distances can be computed using the method described above with slight modifications. The centers of mass \bar{s}_1 and \bar{s}_2 are replaced by the weighted centers of mass and the rotation is computed as if the radii (distance of a point to the weighted center of mass) were scaled according to the weights: $\hat{s}_{i,1} = \sqrt{\omega_i} \cdot (s_{i,1} - \bar{s}_1)$, and $\hat{s}_{i,2} = \sqrt{\omega_i} \cdot (s_{i,2} - \bar{s}_2)$.

Detailed analysis shows that by reorganization of the terms, the explicit computation of the (weighted) centers of mass \bar{s}_1 and \bar{s}_2 can be avoided such that having computed the optimal transformation for the set S_i , the optimal transformation for the set S_{i+1} can be computed in constant time.

Similarity Transformations A similarity transformation (preserving the orientation of closed paths) is a combination of a uniform scaling, a rotation, and a translation. M is the matrix $\begin{pmatrix} c \cdot \cos \varphi & -\sin \varphi \\ \sin \varphi & c \cdot \cos \varphi \end{pmatrix}$ with c being the scaling factor and φ being the angle of rotation around the origin. v is an arbitrary vector. Since there are 4 degrees of freedom, k equals 2. The similarity transformation that minimizes the sum of the weighted squared distances for the sample set $S_n = \{(s_{1,1}, s_{1,2}), \dots, (s_{n,1}, s_{n,2})\}$ can easily be computed by solving the following system of linear equations:

$$\begin{aligned} \sum_{i=1}^n \omega_i \begin{pmatrix} x_{i,1}^2 + y_{i,1}^2 & 0 & x_{i,1} & y_{i,1} \\ 0 & x_{i,1}^2 + y_{i,1}^2 & y_{i,1} & -x_{i,1} \\ x_{i,1} & y_{i,1} & 1 & 0 \\ y_{i,1} & -x_{i,1} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} M[1][1] \\ M[1][2] \\ v[x] \\ v[y] \end{pmatrix} \\ = \sum_{i=1}^n \omega_i \begin{pmatrix} x_{i,2}x_{i,1} + y_{i,2}y_{i,1} \\ x_{i,2}y_{i,1} - y_{i,2}x_{i,1} \\ x_{i,2} \\ y_{i,2} \end{pmatrix}. \end{aligned}$$

This means that for affine transformations not deviating much from similarity transformations (which is assumed for proper mappings of shapes that are perceived similar), the changes in ratios of lengths are rather small which facilitates the augmentation of samples as described in Section 3.2.3.

Reflections A reflection at a line is a transformation for which M is an orthogonal matrix with determinant -1 and therefore it changes the orientation of closed paths. Since a reflection at an arbitrary line can be replaced by a combination of a reflection at a fixed line and a rigid motion, reflections and similarity transformations changing the orientation of closed paths are computed by first reflecting \mathcal{P}_1 at the y -axis (replacing every x -coordinate by its negation) and then computing the optimal rigid motion or similarity transformation respectively.

3.2.5. Weighting Transformations

Besides reducing the impact of individual positional errors, the other main motivation for using augmented samples is the possibility to distinguish between substantial and unsubstantial samples. Two factors are of importance in this context, namely the expressiveness of the sampled parts and the quality of the match. The expressiveness of a sample depends on the geometrical size and on the visual complexity. The weight $\omega(S)$ of a sample set S of cardinality n therefore is determined as the product of three factors, namely a factor $\omega_L(S)$ for the length of the parts, a factor $\omega_V(S)$ for the visual complexity, and a factor $\omega_E(S)$ for the quality of the match. The weight $\omega(S)$ is defined in such a way that all computations building on it are invariant to scaling the input \mathcal{P}_1 or \mathcal{P}_2 , which is desired in most applications.

Depending on the application at hand, the length of the explored part of \mathcal{P}_1 and the length of the explored part of \mathcal{P}_2 may be of different importance. Let $c(S)$ be the prescaling factor for the sample S and let $L_1(S) = (d_f(S) + d_b(S))/c(S)$ and $L_2(S) = (d_f(S) + d_b(S))$ be the lengths of the explored parts of \mathcal{P}_1 and of \mathcal{P}_2 respectively. The idea is to use some kind of weighted geometric mean of the relative lengths:

$$\omega_L(S) := \left(\left(\frac{L_1(S)}{L(\mathcal{P}_1)} \right)^{e_1} \cdot \left(\frac{L_2(S)}{L(\mathcal{P}_2)} \right)^{e_2} \right)^{\frac{1}{e_1+e_2}}$$

with $L(\mathcal{P})$ denoting the total length of a set \mathcal{P} of polylines, and e_1 and e_2 being two parameters for adjusting the relative importances (in the case of complete-complete matching typically $e_1 := e_2 := 1$). Since in the further process decisions are based on linear combinations of the ω_L only, and since

3. Estimation of Shape-Similarity

no value but L_1 and L_2 depends on the sample, the above definition can be replaced by $\omega_L(S) := L_2(S) \cdot c(S)^{-e_1/(e_1+e_2)}$ without changing the results of the process.

The factor $\omega_V(S)$ for the visual complexity is intended to penalize samples that essentially span only one dimension like a single line segment does, compared to samples that really span two dimensions like, e. g., a square does. The spatial distribution of the sample can be described using principal component analysis as introduced in [180]. Let $D(S)$ be the covariance matrix of the coordinates of the sample pairs' second points and let $d_{max}(S)$ be the larger and $d_{min}(S)$ be the smaller of the two eigenvalues of $D(S)$. The quotient d_{min}/d_{max} equals 0 if all points are on a line and it equals 1 if two dimensions are really spanned. In order to get a factor linearly decreasing with increasing 'one-dimensionality' of the sample, $\omega_V(S) := (1 - d_{min}(S)/d_{max}(S)) \cdot \omega_{V,min} + (d_{min}(S)/d_{max}(S)) \cdot 1$ with $\omega_{V,min} < 1$ being the factor of maximal penalization.

The quality of a match clearly depends on the size of the mapping errors relative to the spatial extent of the sample. Still, distortions of small shapes are less obvious to perceive than distortions of large shapes. Therefore also the mapping errors relative to the spatial extent of the whole set of polylines are considered. Let $\bar{\delta}(S) := (1/\omega(S) \cdot \sum_{i=1}^n (\omega_i \cdot \|t_S(s_{i,1}) - s_{i,2}\|^2))^{1/2}$ denote the weighted root mean square error of the sample S . Furthermore let $d_{box}(S)$ be the sum of the side lengths of the smallest axis aligned rectangle containing the sampled points $(s_{1,2}, \dots, s_{n,2})$, and let $d_{img}(\mathcal{P}_2)$ be a measure for the geometrical size⁹ of the polyline set \mathcal{P}_2 . The factor for the quality of the match is then defined as

$$\omega_E(S) := 1 - c_E \cdot \left(\frac{\bar{\delta}(S)}{d_{box}(S)} + \frac{\bar{\delta}(S)}{d_{img}(\mathcal{P}_2)} \right)$$

with c_E being a parameter for adjusting the tolerance against errors.

3.2.6. Clustering Transformations

In the present context, a cluster is a region of limited diameter in transformation space subsuming a considerable amount of weight of the enclosed transformations. It is assumed that clusters with large total weight correspond to transformations that map large parts of one polyline set to corresponding parts in the other polyline set (see also [201, p 21]).

⁹ In the context of retrieval of figurative images this value should be defined based on the size of the image, e. g., as proposed in Section 4.2.

Basics Since in general a whole set of transformations approximately match a set of sample pairs' first elements to the corresponding second elements, ideally a transformation contained in the maximum number of such sets¹⁰ should be computed. For translations and rigid motions there has been work in this direction [40], however, due to the high computational complexity, the given algorithms are not applicable in the given context.

Most techniques proposed for clustering transformations, on the other hand, are based on assigning a single transformation to a sample, partitioning transformation space (typically independently partitioning every parameter uniformly), and histogramming the transformations over the cells of the partition [173]. These methods, however, completely discard the effects of the transformations on the objects that are transformed.

Applied to a shape near the origin, two rotations may yield nearly the same results, whereas applied to a shape far from the origin the same two rotations may yield very different results (see Figure 3.5 for an example). Clustering two given transformations may be absolutely reasonable for some inputs whereas it might be inappropriate for others. The same problem may occur when scalings (or other linear transformations) are involved.

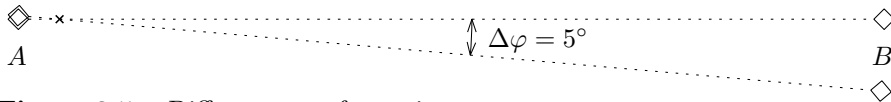


Figure 3.5. Different transformations:
Shapes that are transformed by two rotations which differ only by 5° . The images of shape A are almost the same, whereas the images of shape B strongly differ.¹¹

Normalizing all input sets of polylines such that they are centered at the origin would be one possible way to tackle this problem for affine transformations, however, defining a distance measure on transformations based on their effects on the transformed objects, allows to develop clustering algorithms that are completely independent of the actual class of transformation at hand.

¹⁰ In the weighted case analogously a transformation contained in an intersection of sets such that the sum of weights is maximal.

¹¹ Please note that after applying an additional translation, the situation may also be the other way round.

3. Estimation of Shape-Similarity

Given a set \mathcal{P} of polylines (or in general, shapes in the plane), let \mathcal{P}^{box} be the set consisting of the 4 corner points of the smallest axis aligned rectangle containing \mathcal{P} (the bounding box). A distance measure on a class \mathcal{T} of transformations can then be defined as $d_{\mathcal{P}}(t_1, t_2) := \max_{p \in \mathcal{P}^{box}} \{\|t_1(p) - t_2(p)\|\}$. It is easy to see that $d_{\mathcal{P}}$ satisfies the triangle inequality which will be utilized in the clustering algorithm described in the following.¹²

Since in the given context the diameter of any reasonable cluster of transformations is limited, the standard methods for clustering based on distance information like, for instance k -means clustering [154] cannot be used. Complete linkage clustering [58] possibly could yield good results. However, due to the worst case time complexity $\Theta(n^2 \cdot \log(n))$ for clustering n transformations, the given algorithm is also not favorable here.

Idea of the Proposed Approach Let $T = \{t_1, \dots, t_n\}$ be a set of transformations intended to transform the set \mathcal{P} of polylines, and let $\omega(t_i)$ denote the weight of transformation t_i . For a fixed cluster radius r a cluster $C(t_i)$ with center $t_i \in T$ is defined as the set of transformations with distance less than r to the center: $C(t_i) = \{t_j \in T \mid d_{\mathcal{P}}(t_j, t_i) < r\}$. The weight of a cluster is then defined as the sum of the weights of its elements. This definition is closely related to what is called *naive density estimator* in statistics [209]. The transformations that are considered as cluster centers are identified as follows: $t_i \in T$ is called *dominator* of t_j if and only if $d_{\mathcal{P}}(t_i, t_j) \leq r$, $\omega(t_i) > \omega(t_j)$, and no other transformation is dominator of t_i . Each transformation $t \in T$ that has no dominator is the center of a cluster C_t . In other words: a transformation t either is the center of a cluster or it is contained in at least one cluster of its dominators (see Figure 3.6 for an illustration).

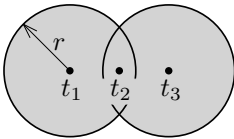


Figure 3.6. Definition of cluster centers:

Let $\omega(t_1) > \omega(t_2) > \omega(t_3)$, then t_1 is dominator of t_2 and therefore t_2 is *not* dominator of t_3 . Since neither t_1 nor t_3 have a dominator, each of them defines a cluster and t_2 is contained in both of these clusters.

The clusters may be determined by iteratively taking the transformation with highest weight as center of a cluster, removing the cluster's members from the set of potential centers and continuing with the reduced set. A naive algorithm

¹² Given that the points of \mathcal{P}^{box} are not collinear, $d_{\mathcal{P}}$ also satisfies the other metric conditions for the class of affine transformations. For more general classes of transformations the *isolation* condition might be violated.

would need time quadratic in the number of transformation, but this can be decreased by partitioning transformation space based on the distances $d_{\mathcal{P}}$ and organizing the partition in a rooted, ordered tree which holds the clusters.

Partitioning Transformation Space Let r be the cluster radius and let h be the depth of the tree. A node u on level k represents a $d_{\mathcal{P}}$ -ball $b(u)$ with radius $r_k = 2^{h-k} \cdot r$ around the center of the cluster that is stored in u . Every node u on a level $< h$ may have arbitrarily many children, each of which represents a ball of half the radius, centered inside $b(u)$. The root node represents a ball with radius r_0 such that all transformations from T lie inside this ball.

Since balls from the same levels as well as balls from different levels may intersect, a hierarchy is defined as follows: A node u is responsible for that part of transformation space its parent was responsible for, which lies inside its own $d_{\mathcal{P}}$ -ball and does *not* lie inside one of the $d_{\mathcal{P}}$ -balls of its predecing siblings (Figure 3.7 shows an example). Formally, given a node u on level l , let $(v_0, v_1, \dots, v_l = u)$ be the sequence of nodes on the path from the root to u . Furthermore let $c_i(v)$ denote the i -th child of a node v and let (i_1, \dots, i_l) be the numbers such that v_j is the i_j -th child of v_{j-1} . Then the part of transformation space that u is responsible for is

$$b'(u) = \bigcap_{k=1}^l \left(b(v_k) \setminus \bigcup_{j=1}^{i_k-1} b(c_j(v_{k-1})) \right).$$

The responsibility region of a node only refers to the centers of the clusters stored in the node's subtree, but the clusters themselves may protrude over these responsibility regions by the cluster radius r . In an r -neighborhood of the boundary between two responsibility regions, a transformation may therefore belong to clusters stored in two different subtrees. However, given a node u on level k , representing a $d_{\mathcal{P}}$ -ball $b(u)$ with radius r_k around center t_u , for any transformation t' such that $d_{\mathcal{P}}(t_u, t') < r_k - r$ all the clusters containing t' *must not* be centered inside responsibility region of a succeeding sibling of u , and for any transformation t'' such that $d_{\mathcal{P}}(t_u, t'') > r_k + r$ all the clusters containing t'' *must not* be centered inside the responsibility region of u .

In Figure 3.7, when searching for the clusters containing transformation t , the subtree of node $c_1(v_0)$ has to be considered but the search cannot be restricted to it, because $r_1 - r < d_{\mathcal{P}}(c_1(v_0), t) < r_1 + r$. The subtrees of the predecing siblings $c_2(v_0)$ and $c_3(v_0) = v_1$ can be excluded because the distance of t to the respective centers exceeds $r_1 + r$. Since $d_{\mathcal{P}}(c_4(v_0), t) < r_1 - r < r_1 + r$, the subtree of node $c_4(v_0)$ again has to be considered, but no further sibling has to be looked at.

3. Estimation of Shape-Similarity

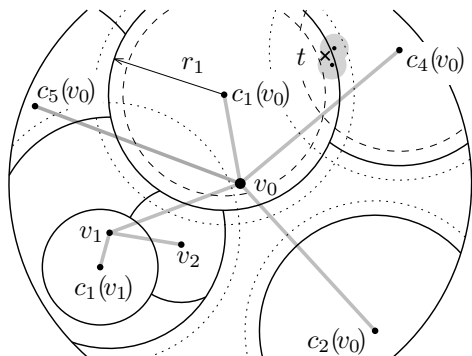


Figure 3.7. Partition of transformation space: example showing responsibility regions (bounded by continuous arcs) relevant for a path $(v_0, v_1 = c_3(v_0), v_2 = c_2(v_1))$, plus a transformation t (marked by \times) that belongs to clusters (gray) from different subtrees, relevant outer boundaries (dotted), and inner boundaries (dashed).

Algorithm Let (t_1, \dots, t_n) be the transformations sorted such that their weights are in decreasing order. The clustering is then done in two phases. In the first phase the cluster centers are identified and the tree representing the partition of transformation space is constructed. Given the tree for (t_1, \dots, t_{i-1}) , whether t_i constitutes a new cluster (whether it has no dominator) can easily be determined by searching the tree for a cluster covering t . If no such cluster exists, a new cluster with center t_i is created and inserted into the tree. In the second phase all the transformations of T are assigned to the clusters. In order to do so, for every $t \in T$ the final tree has to be searched for all clusters covering t . The tree, therefore, needs to provide two functionalities, namely searching for all clusters covering a transformation (or deciding whether such a cluster exists) and inserting a cluster into the tree.

A (sub-)tree rooted at a node u can be searched for all clusters covering a transformation t using Algorithm 3.2. If the cluster stored in u covers t , then this cluster is contained in the return set (in the decision version the algorithm may terminate then), and sequentially every child of u is tested whether it also has to be searched and whether the succeeding siblings can be dropped.

Given a node u with ball b_u covering a transformation t , a cluster with center t can be inserted into the (sub-)tree rooted in u using Algorithm 3.3. If there exists at least one child of u , representing a ball covering t the cluster is recursively inserted into the first such child. Otherwise, a new child with center t is created and appended to the list of child nodes.

If the ball of the current root node does not cover t , the root may be made child of a new root node with doubled radius until this new root's ball does cover t .

Algorithm 3.2: Search subtree of node u for clusters covering transformation t

```

 $U \leftarrow (u)$  /*nodes to search*/,  $C \leftarrow \emptyset$  /*clusters found*/
while  $U \neq \emptyset$  do
   $v \leftarrow$  extract first element of  $U$ 
  if  $d_{\mathcal{P}}(v, t) \leq r$  then
    | add  $v$  to  $C$ 
  end
   $r_v \leftarrow$  radius of  $v$ ,  $j \leftarrow$  number of children of  $v$ 
  for  $i = 1$  to  $j$  do
    | if  $d_{\mathcal{P}}(c_i(v), t) \leq r_v + r$  then
      | | add  $c_i(v)$  to  $U$ 
    | end
    | if  $d_{\mathcal{P}}(c_i(v), t) < r_v - r$  then
      | | break
    | end
  end
end
return  $C$ 

```

Algorithm 3.3: Insert cluster with center t into subtree of node u

```

 $v \leftarrow u$  /*current node*/
while true do
   $r_v \leftarrow$  radius of  $v$ ,  $j \leftarrow$  number of children of  $v$ 
  for  $i = 1$  to  $j$  do
    | if  $d_{\mathcal{P}}(c_i(v), t) \leq r_v/2$  then
      | |  $v \leftarrow c_i(v)$ , continue with outer loop
    | end
  end /*no responsible child found*/
  create new child  $c_{j+1}(v)$  with center  $t$  storing the cluster
  break loop
end

```

3.2.7. Analysis

Generating Transformations Let n_1 be the number of vertices of \mathcal{P}_1 and let n_2 be the number of vertices of \mathcal{P}_2 . Using binary search on an array of accumulated weights of the vertices, a random initial sample can be drawn in time $O(\log(n_1) + \log(n_2))$. Since every step of the augmentation of the sample (determining next point pair, computing the transformation, and checking consistency) can be done in constant time, a vote can be generated in time $O(n_1 + n_2)$.

In experiments with real-world data¹³, the average number of augmentations was significantly smaller than the total number of vertices. Regression analysis resulted in $9.7 + 0.09 \cdot (n_1 + n_2)$ for the *MPEG 7* data set (one polyline per image) and $7.0 + 0.004 \cdot (n_1 + n_2)$ for the *UK trademarks* set (multiple polylines per image).

Clustering Transformations Let d_1^{box}, d_2^{box} denote the diameters of the bounding boxes of shape sets \mathcal{P}_1 and \mathcal{P}_2 respectively. Any transformation t generated by the random experiments will fulfill the condition that the transformed bounding box of \mathcal{P}_1 at least touches the bounding box of \mathcal{P}_2 . Given the definition of $d_{\mathcal{P}_1}$, it is therefore easy to see that for translations and rigid motions, $d_{\mathcal{P}_1}(t_i, t_j) \leq d_2^{box} + 2 \cdot d_1^{box}$ for any pair of transformations t_i, t_j . With r being the cluster radius, the maximum depth of the tree is then bounded from above by $\lceil \log((d_2^{box} + 2 \cdot d_1^{box})/r) \rceil$. Since they are not length preserving, for similarity transformations and general affine transformations, the space is not bounded in such a natural way. However, if the application provides bounds on the maximum scaling factor c_{max} and on the maximum deviation dev_{max} from a similarity transformation, the distance between any pair of transformations can be bounded based on these values: $d_{\mathcal{P}_1}(t_i, t_j) \leq d_2^{box} + 2 \cdot d_1^{box} \cdot c_{max} \cdot dev_{max}$ for any pair of transformations t_i, t_j . The maximum depth of the tree is then bounded from above by $\lceil \log((d_2^{box} + 2 \cdot d_1^{box} \cdot c_{max} \cdot dev_{max})/r) \rceil$.

Given a bounded set \mathcal{X} and a distance measure d , a set $\mathcal{Y} \subset \mathcal{X}$ is called an ε -packing if and only if $\forall y_1, y_2 \in \mathcal{Y} : d(y_1, y_2) > 2\varepsilon$. The size of the largest ε -packing is called the packing number $P(\mathcal{X}, \varepsilon)$. For an m -dimensional ball b with radius r , the packing number $P(b, r/4)$ is in $O(8^m)$. In particular, for m being a constant, the packing number $P(b, r/4)$ is also constant (see [54]). This observation can directly be applied to bound the maximal degree of a node in the tree for \mathcal{T} being the class of translations, since the children of a node u define a Euclidean $r_u/4$ -packing of the ball $b(u)$.

¹³ The mapping algorithm has been applied to 1 000 randomly chosen pairs of images from the *MPEG 7* data set and to 1 200 pairs (50 randomly chosen images for each query) of images from the *UK trademarks* set, using similarity transformations.

For transformations with more than 2 degrees of freedom, the bound on the maximal degree of a node in the tree as derived from analog reasoning may even be improved by exploiting the definition of $d_{\mathcal{P}}$. Since $d_{\mathcal{P}}$ is defined as the maximum of the distances of 4 point-pairs, every packing with respect to $d_{\mathcal{P}}$ in transformation space having cardinality P' , induces 4 packings with respect to the Euclidean distance in the \mathbb{R}^2 , having a total cardinality of at least P' .¹⁴ The maximal degree of a node in the tree is therefore bounded by $4 \cdot O(8^2)$ which obviously is constant.

For transformations in the neighborhood of a responsibility region's boundary more than one node of a level may have to be searched. Therefore, the nodes traversed during a search do not necessarily form a path, but may form a tree (in the following called *searching tree*). Straightforward attempts to bound the complexity of the searching trees based on packing arguments do not lead to convenient results. However, in experiments with real-world data¹⁵, the average degree of non-leaves in the searching trees was smaller than 1.2 and the number of nodes in a searching tree was smaller than 1.1 times the depth of the original tree on the average. Assuming the average complexity of the searching tree to be bounded by a constant times the depth of the original tree, the running time of clustering n transformations is in $O(n \cdot \log(n) + n \cdot \log((d_2^{box} + 2 \cdot d_1^{box} \cdot c_{max} \cdot dev_{max})/r))$.

¹⁴ Each point in transformation space corresponds to 4 points in Euclidean space (1 for every element of \mathcal{P}^{box}), and each element of a packing in transformation space has to contribute to at least 1 of the 4 packings in the \mathbb{R}^2 .

¹⁵ The mapping algorithm has been applied to 1 000 randomly chosen pairs of images from the *MPEG 7* data set using similarity transformations.

3.3. Similarity Estimation

The aim of this section is to find a measure of similarity between sets of polylines representing shapes in figurative images, that complies with the perceived similarity of these images.

3.3.1. Related Work

Since estimating the similarity of shapes is an important task in a wide variety of applications, there is also a wide variety of definitions of (dis-)similarity measures. Given a specific application, whether a (dis-)similarity measure is worth considering may depend on two very basic issues: First, whether it can be applied to pairs of arbitrary sets of shapes or to pairs of single shapes only. Second, whether it is invariant with respect to asynchronous transformation under the classes of translations, rigid motions, etc. or not. In the following an overview over some very different approaches is given.

Geometry-Based Measures Probably the most well known geometry-based measure of dissimilarity is the *Hausdorff distance* d_H (see Section 1.3.4). It may be used for arbitrary sets of shapes by applying it either to the interiors of the shapes or to the boundaries of the shapes. In order to reduce the sensitivity to noise, several variants of the Hausdorff distance have been proposed, such as the *partial Hausdorff distance* [119] (also called *percentile based Hausdorff distance*) and the *mean Hausdorff distance* [67]—both of which do actually not fulfill the metric properties. Other measures include the *Fréchet distance* d_F (see Section 1.3.4) which, however, is only defined for pairs of single shapes, and the *area of symmetric difference* which is defined as $\|(S_1 \cup S_2) \setminus (S_1 \cap S_2)\|$ and may be applied to pairs of arbitrary sets of region shapes. These measures have a simple and sound mathematical basis, but show only limited conformance with perceived similarity (see, e.g., Figure 1.11 on page 50). In [103] a distance measure has been introduced which is based on differences in the area of visibility regions. This measure may be applied to arbitrary sets of (boundaries of) region shapes, but it is not invariant with respect to asynchronous transformations. It fulfills the metric axioms and moreover, is robust against noise, crack, blur and deformation.

Statistical Measures Statistical measures are based on the representation of a shape (or a set of shapes, respectively) by a high-dimensional vector (or point in high-dimensional space), each dimension standing for some aspect of the shapes. Once the vectors have been determined, two shapes S_1 and S_2 (or sets of shapes \mathcal{S}_1 and \mathcal{S}_2 respectively) are then compared by applying some

measure of (dis-)similarity on the corresponding vectors v_1 and v_2 (see also Section 1.3.4 page 51). Possible measures are, e. g., the Minkowski distance, the dynamic partial function (for both see Section 1.3.4), the *cosine similarity* $\langle v_1, v_2 \rangle / (\|v_1\| \cdot \|v_2\|)$, and the Mahalanobis distance (see, for instance [216]).

Given a polygonal region shape S with boundary P , some of the quantities that may be used to describe S are listed in the following (taken from [10], for a more comprehensive list see also [225]):

- circularity: $4 \cdot \pi \cdot A / L^2$
- right-angleness: r/n
- sharpness: $\frac{1}{n} \sum \max(0, 1 - (2 \cdot |\Theta_i - \pi| / \pi)^2)$
- complexity: $10^{-(7/n)}$
- aspect ratio: l/w
- stuffedness: A/A_R
- ...

with A denoting the area of S , L the length of P , n the number of vertices of P , r the number of almost right angles between consecutive edges of P , Θ_i the angle between the the i -th and the $i+1$ -th edge of P , l the length, w the width, and A_R the minimum area of any rectangle enclosing S .

Apart from the questionable use of distance measures fulfilling the triangle inequality (see Section 1.3.3.3 page 42 for a discussion) there are also other issues that heighten the risk of rating perceptually similar shapes dissimilar: Shapes that are perceived as being very similar may—depending on the actual bounding polylines—heavily differ in such quantities as, e. g., complexity and right-angleness. Moreover, different groupings of shapes (see Section 2.5 and Figure 2.24 on page 110) may lead to completely different representations.

Moment-Based Measures Given a two-dimensional function $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ and two integers p and q , the (*raw*) *moment* is defined as

$$M_{p,q}^r(f) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \cdot f(x, y) dx dy.$$

By using the characteristic function of a shape S (or a set of shapes \mathcal{S} , respectively), its moment is defined analogously as

$$M_{p,q}^r(S) := \iint_S x^p y^q dx dy.$$

3. Estimation of Shape-Similarity

Let $\bar{x} = M_{1,0}^r(S)/M_{0,0}^r(S)$ and $\bar{y} = M_{0,1}^r(S)/M_{0,0}^r(S)$ be the coordinates of the centroid of S , then the *central moment* is defined as

$$M_{p,q}^c(S) := \iint_S (x - \bar{x})^p (y - \bar{y})^q dx dy.$$

Some combinations of these moments may be used to define shape descriptors that are invariant to translation, scaling, and rotation (see, for instance [245]) and even broader sets of geometric deformations have been considered [88]. In the context of shape retrieval also moments based on *Zernike polynomials* as defined in [251] are used (see, e. g., [23], [130], and [238]).

Moment-based descriptions of shapes are very commonly used in the retrieval of figurative images (see Section 4.1). However, different depictions of shapes (e. g., by a region or by the outline—see Figure 1.2 on page 22) may lead to completely different representations when the moments are computed based on pixel intensities. In addition, perceptually less important shapes such as frames may dominate the representation by moments and may lead to unwanted results.

Edge (Direction) Histograms The idea behind *edge histograms* is to partition the plane—usually by a regular grid either in Cartesian coordinates or in polar coordinates—and to measure the length of the part of the shape boundary¹⁶ in each cell. The information about the spatial distribution of the shape boundary can then be further processed to obtain shape representations that are invariant to translation, scaling, and rotation and that can be used for shape retrieval (see, for instance [42]).

For *edge direction histograms* on the other hand, not a partition of the plane, but a partition of the possible directions of the shape boundary is used (see, for example [52]). In order to get richer descriptions of the shapes/images, also combinations of edge direction histograms with edge histograms [183] and combinations with information about the neighborhood relations between the edges [159] have been considered.

Edge (direction) histograms are almost robust with respect to differences in the depiction of shapes and to differences in the actual grouping of shapes. However, as a consequence of the partitioning translations and rotations may completely change the representation of a shape or image. Normalization on

¹⁶ Most of the work concerning edge histograms or edge direction histograms actually considers arbitrary sets of edges detected in raster graphics and, instead of measuring lengths, just counts edge pixels. However, the approaches may also be applied to explicitly given shapes.

the other hand entails the risk of different representations for similar shapes or images due to perceptually irrelevant additional features (e. g., noise, frames, etc.).

Curvature-, Turning-, and Signature Functions A curve C is usually specified by an explicit or implicit description of the points belonging to C . However, given a reference point o on C and a reference direction v , the curve is also completely determined by the curvature as a function of the arc-length, as well as by the angle between the tangent and v as a function of the arc-length. This fact has been exploited for shape recognition and similarity estimation: For example, in [241] the curvature functions were used to derive sequences of descriptions of discrete points, which were then compared using string matching algorithms. In [14] some measures of dissimilarity for polygonal curves were defined based on the turning function and arbitrary distance measures in function space. Moreover, also other definitions of functions have been proposed for similarity estimation. For example in [175] a so called *signature function* assigning every point p on the curve the length of the part of the curve left to the tangent in p was used. One major drawback of this approach is that it cannot be used to distinguish between different convex shapes, since in this case the signature functions equal 1 everywhere.

Curvature Scale Space The main idea behind shape matching based on *curvature scale space* (cf. [166]) is to trace the zero-crossings of the boundaries' curvature while the shapes are successively blurred. Let C be a closed boundary curve given in parametric representation $C(s) = (x(s), y(s))$ with respect to the arclength s , and let $\kappa(s)$ be the signed curvature. The zero-crossings of κ are the values of s such that the corresponding points on C separate convex from concave pieces of the boundary. Applying one-dimensional Gaussian kernels $G(s, \sigma)$ with increasing standard deviation σ (scale) on C causes the zero-crossings to move continuously, and pairs of them to meet and vanish. A measure of shape similarity can be derived from a limited number of (s, σ) -pairs where zero-crossings of the curvature vanish—namely the ones with largest scale σ .

This approach is invariant with respect to asynchronous transformation under the classes of translations, rotations, and scalings, however, it may only be applied to pairs of single shapes. Moreover, it also suffers from the fact that convex shapes (as the curvature does not change the sign) cannot be distinguished between.

3. Estimation of Shape-Similarity

Skeletons Skeletons can be thought of as thinned versions of the shape. Probably the most well known type of skeletons is the *medial axis* as introduced in [31]. It is defined as the locus of the centers of circles that are bitangent (from the inside) to the boundary of the curve. In association with the function that assigns every point of the medial axis the radius of the corresponding circle, it is called the *medial axis transform* which completely describes the shape. Also other variants of skeletons have been introduced, as for instance the *straight line skeleton* [4] and the closely related *linear skeleton* [215].

These skeletons are usually not directly used for similarity estimation of polygonal shapes, because they are very sensitive to the occurrence of convex vertices (see Figure 2.10 on page 84 for an example). More suitable representations of the relevant features can be achieved by *skeleton pruning* (see [18] for an overview), by determining a hierarchy on the skeleton features (see, e. g., [176]), or by building the so called *shock graph* as defined in [208].

The nodes of the shock graph are pieces of the medial axis. Four types of such pieces are distinguished: (1) maximal connected linear¹⁷ pieces where the radii of the corresponding circles increase monotonically, (2) points where the radius reaches a local minimum, (3) maximal connected linear pieces where the radius is (almost) constant, and (4) points where the radius reaches a local maximum. Starting from the vertices corresponding to local maxima of the radius, edges are inserted directing to vertices corresponding to neighboring pieces with smaller radius. For examples on how to derive a measure of shape similarity based on these shock graphs see [208] and [204].

Skeleton pruning and using shock graphs reduce the dependence on the local properties of the shapes' boundaries. However, shapes perceived very similar may have skeletons that also differ fundamentally (see Figure 3.2 (c), (d), and (e) for an example) and this limits the suitability of skeleton-based approaches especially for abstract geometric shapes.

3.3.2. Idea of the Proposed Approach

In order to be applicable in the context of perceived similarity, a similarity measure on shape sets should be robust with respect to different representations and groupings of the shapes (see Figure 1.11 on page 50 and Figure 2.24 on page 110), but at the same time it should take account of the local geometry of the shapes (see Figure 1.11). The basic idea applied here is to use a resemblance function ϕ that is defined on the boundaries of the shapes and that assigns to every boundary point belonging to one shape set S_1 a value of how good

¹⁷ Here, *linear* means that no branch point (point where the corresponding circle touches the boundary in more than 2 points) is included.

it is represented by the other shape set S_2 . For rating how good a point is represented, 2 perceptual factors are incorporated, namely proximity and parallelism (that is, analogy in the courses of the boundaries). Deriving a value of similarity from the integrals of the resemblance functions leads to measures that are almost robust with respect to different representations, to crack, and to noise.

3.3.3. Definition of the Similarity Measure

For a set \mathcal{P} of polylines, let $E(\mathcal{P})$ denote the union of the sets of edges of the polylines in \mathcal{P} . Given two sets \mathcal{P}_1 and \mathcal{P}_2 of polylines, let g be a straight line segment from $E(\mathcal{P}_1)$ with endpoints p_0 and $p_0 + \vec{v}$, let h be a non-orthogonal segment of $E(\mathcal{P}_2)$ with endpoints q_1, q_2 , and let g' and h' denote the supporting lines of the segments g and h respectively. Furthermore, let l_g be the length of g .

The resemblance function takes two things into account, namely proximity and analogy in slope. For evaluating the proximity, to every point of g a value describing the distance to h is assigned as follows: For every $\lambda \in \mathbb{R}$ let $p(\lambda) := p_0 + \lambda \cdot \vec{v}$ and let $q(\lambda)$ be the point on h' such that its orthogonal projection onto g is exactly $p(\lambda)$. Let furthermore λ_1 be the value such that $q(\lambda_1) = q_1$ and λ_2 be the value such that $q(\lambda_2) = q_2$ (without loss of generality $\lambda_1 < \lambda_2$). Figure 3.8 shows an illustration.

A function $d_{g,h}(\lambda)$ that describes the distance¹⁸ of a point on g to the segment h and that is—unlike the minimum Euclidean distance from $p(\lambda)$ to a point of h —piecewise linear in λ , can be defined as

$$d_{g,h}(\lambda) := \begin{cases} \|p(\lambda) - p(\lambda_1)\| + \|p(\lambda_1) - q_1\|, & 0 \leq \lambda < \lambda_1 \\ \|p(\lambda) - q(\lambda)\|, & \lambda_1 \leq \lambda \leq \lambda_2 \\ \|p(\lambda) - p(\lambda_2)\| + \|p(\lambda_2) - q_2\|, & \lambda_2 < \lambda \leq 1. \end{cases}$$

Since proximity and distance are inversely related, the values of $d_{g,h}(\lambda)$ have to be converted (see Section 1.3.2 for general considerations). In [185] proximity of shape boundaries was rated by an exponentially decreasing *inverse distance function*. However, there it was intended to push solutions towards an exact alignment in an optimization process. In the given context small deviations in position should not result in an excessive decrease of the resemblance function but instead should be tolerated. Therefore, the absolute value of the derivative of the function converting distance to proximity should be

¹⁸ Please note that since $d(\lambda)$ describes the distance between a point and a segment, it *cannot* be a *distance measure* in the sense that it fulfills the metric properties.

3. Estimation of Shape-Similarity

small for small values. Given a threshold d_{max} on the distance, a measure of proximity that conforms to this demand can be defined as $\alpha_{g,h}(\lambda) := \max(1 - (d_{g,h}(\lambda)/d_{max})^2, 0)$. Figure 3.8 shows the graph of the conversion function.

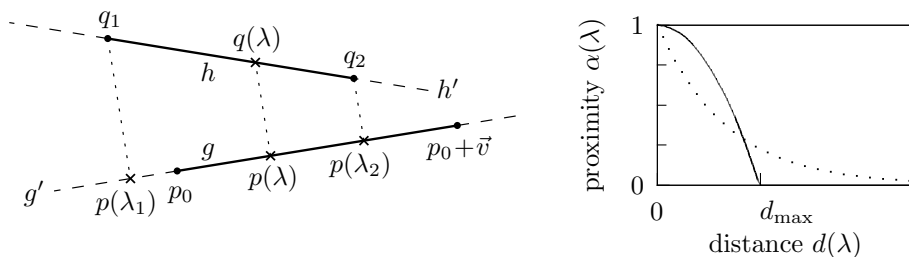


Figure 3.8. Distance function and proximity function: notations used in the definition of $d(\lambda)$, plus conversion function used for the definition of $\alpha(\lambda)$ (continuous), and some exponentially decreasing conversion function (dotted).

Apart from proximity, the resemblance of two line segments also depends on their slopes. In order to take this into account, a slope factor is introduced as $\beta_{g,h} := \cos^6(\angle(g, h))$. This definition ensures that pairs of lines that look almost parallel get high values (greater 0.9 for differences in slope less than 10°) and pairs with very different slope get low values (below 0.15 for differences in slope greater than 45°).

The *resemblance function* ϕ_{g,\mathcal{P}_2} for a line segment $g \in E(\mathcal{P}_1)$ is defined as a combination of the proximity function and the slope factor: $\phi_{g,\mathcal{P}_2}(\lambda) := \max_{h \in E(\mathcal{P}_2)} (\alpha_{g,h}(\lambda) \cdot \beta_{g,h})$. With the given definitions of d and α , the function ϕ is piecewise quadratic.

A directed measure of resemblance, rating how good the set \mathcal{P}_1 is represented by \mathcal{P}_2 can then simply be defined as

$$\Phi^-(\mathcal{P}_1, \mathcal{P}_2) := \frac{\sum_{g \in E(\mathcal{P}_1)} \left(l_g \cdot \int_0^1 \phi_{g,\mathcal{P}_2}(\lambda) d\lambda \right)}{\sum_{g \in E(\mathcal{P}_1)} l_g}.$$

Depending on the application at hand, several measures of similarity may be derived from this directed measure of resemblance. For two sets \mathcal{P}_i and \mathcal{P}_j of polylines, let $\mathcal{P}_j^{(i)}$ be the parts of \mathcal{P}_j that are located inside a limited region around \mathcal{P}_i .¹⁹

All measures based on Φ^- will be subsumed under the name *substitution similarity*. The measures used throughout this work are defined as follows:

- The symmetric *complete-complete* measure
 $\Phi^{cc}(\mathcal{P}_1, \mathcal{P}_2) := 0.5 \cdot (\Phi^-(\mathcal{P}_1, \mathcal{P}_2) + \Phi^-(\mathcal{P}_2, \mathcal{P}_1))$
 may be used to rate how similar 2 sets of shapes are as a whole.
- The *complete-partial* measure
 $\Phi^{cp}(\mathcal{P}_1, \mathcal{P}_2) := 0.5 \cdot (\Phi^-(\mathcal{P}_1, \mathcal{P}_2) + \Phi^-(\mathcal{P}_2^{(1)}, \mathcal{P}_1))$
 may be used to rate how similar the set \mathcal{P}_1 is to a part of \mathcal{P}_2 , totally ignoring additional parts of \mathcal{P}_2 .
- The *complete-semi-partial* measure
 $\Phi^{cs}(\mathcal{P}_1, \mathcal{P}_2) := 0.25 \cdot (2 \cdot \Phi^-(\mathcal{P}_1, \mathcal{P}_2) + \Phi^-(\mathcal{P}_2, \mathcal{P}_1) + \Phi^-(\mathcal{P}_2^{(1)}, \mathcal{P}_1))$
 may be used to rate how similar the set \mathcal{P}_1 is to a part of \mathcal{P}_2 , however, not totally ignoring additional parts of \mathcal{P}_2 .

Furthermore, from these similarity measures corresponding dissimilarity measures may be derived. However, in general they will not fulfill the triangle inequality.

3.3.4. Computation of the Similarity Measure

Algorithm 3.4 is a straight forward implementation for computing $\Phi^-(\mathcal{P}_1, \mathcal{P}_2)$.

Algorithm 3.4:

```

 $\phi_{sum} \leftarrow 0$  /*the integral*/,  $l_{sum} \leftarrow 0$  /*the length*/
forall the  $g \in E(\mathcal{P}_1)$  do
  | forall the  $h \in E(\mathcal{P}_2)$  do
  | | compute  $\phi_{g,h}$ 
  | end
  |  $\phi_{g,\mathcal{P}_2} \leftarrow$  upper envelope of  $\phi_{g,h_1}, \dots, \phi_{g,h_m}$ 
  |  $\phi_{sum} \leftarrow \phi_{sum} + l_g \cdot \int \phi_{g,\mathcal{P}_2}$ ,  $l_{sum} \leftarrow l_{sum} + l_g$ 
end
return  $\phi_{sum}/l_{sum}$ 

```

Since the resemblance function is piecewise quadratic, it can easily be represented by lists of 4-tuples (3 coefficients, 1 boundary). The integrals for every such piece may be computed directly.

¹⁹ Here, $\mathcal{P}_j^{(i)}$ is defined based on the bounding box of \mathcal{P}_i : Let B_i be the smallest axis aligned rectangle containing \mathcal{P}_i and let B'_i be the rectangle obtained from enlarging B_i by the factor 1.2, then $\mathcal{P}_j^{(i)} := \mathcal{P}_j \cap B'_i$.

3. Estimation of Shape-Similarity

3.3.5. Analysis

Let n be the number of edges in $E(\mathcal{P}_1)$, and m be the number of edges in $E(\mathcal{P}_2)$. For a given segment $g \in E(\mathcal{P}_1)$ every $\phi_{g,h}$ consists of at most 3 non-zero quadratic pieces. Therefore ϕ_{g,\mathcal{P}_2} is the upper envelope of at most $3m+1$ pieces, each pair of them (unless equal) intersecting at most twice. According to the upper bound on the length of Davenport-Schinzel sequences the complexity of the upper envelope of these $3m+1$ pieces is bounded by $O(m \cdot 2^{\alpha(m)})$ with α being the inverse Ackermann function [1]. Using a divide and conquer algorithm this upper envelope can be computed in time $O(m \cdot \alpha(m) \cdot \log(m))$ [108]. Every piece can be constructed and integrated in constant time, the overall running time of the algorithm therefore is in $O(n \cdot m \cdot \alpha(m) \cdot \log(m))$.

This analysis does not make use of the fact that the distances are derived from line segments in the plane. However, please note that due to the influence of the slope, two-dimensional Voronoi-diagrams may *not* be used to determine the segment yielding the best resemblance value for a point p . Figure 3.9 shows an example where the total complexity of the upper envelopes is in fact quadratic in the number of segments.

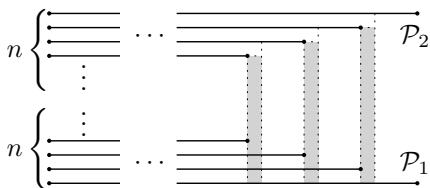


Figure 3.9. Quadratic complexity of upper envelope: scopes of the end points (gray) and of the interiors alternate $\Omega(n)$ times.

3.3.6. Properties

Let $\delta^{cc} := 1 - \Phi^{cc}$ be the measure of dissimilarity derived from Φ^{cc} . δ^{cc} is normalized and it is a semimetric (in other words, it fulfills the properties *non-negativity*, *small self-distance*, *isolation*, and *symmetry*). However, it does not fulfill the triangle inequality. δ^{cc} is invariant with respect to synchronous transformation under the classes of translations, rotations, reflections, and—if d_{max} is chosen relative to the geometric extent of the input—also scalings.

With respect to the formal definitions of robustness, δ^{cc} is crack robust, but *not* deformation, blur, and noise robust (see Figure 3.10). However, in practice reasonable deformations and noise will only cause small changes of the results. Moreover, violations of the noise robustness can only decrease but not increase

the dissimilarity value. This means that the addition of noise will not cause similar sets of shape to become dissimilar. δ^{cc} is distributive, but not monotone and with respect to the formal definition it is also not sensitive.

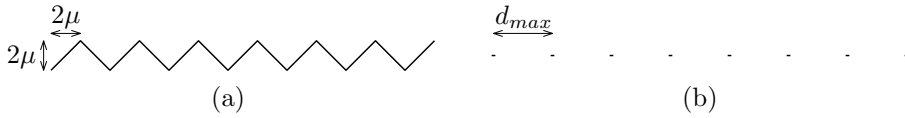


Figure 3.10. Counterexamples for deformation- and noise robustness: (a) deformation of a straight line segment that, when comparing to a straight line segment, causes a difference of δ^{cc} which is greater than 0.85, (b) “noise” that, when comparing to a straight line segment, causes a difference of δ^{cc} which is greater than 0.125.

Applying δ^{cc} to the contours detected in figurative images yields a measure of dissimilarity on these images which is invariant with respect to changes of the colors (as long as they are distinct), in particular to an inversion of black and white. Moreover, also figurative images depicting shapes in different ways (by regions or by outlines) can be compared adequately.

3.4. Evaluation

In order to demonstrate its applicability to a broad range of problems, the *substitution similarity* has been tested on three very different sets of data: a set of Chinese characters, the *MPEG 7* data set (see Section 1.6.2.1), and the *UK trademarks* set (see Section 1.6.2.2).

Chinese Characters In [50], several distance measures for geometric graphs were tested on a set of Chinese glyphs, namely 4 176 abstract Chinese characters each one written in 6 different fonts. From each glyph a graph essentially representing the basic strokes was extracted by simplifying its medial axis. One font was used as reference and each of the remaining $5 \times 4\,176$ graphs was queried for: The distance of each of the 4 176 reference graphs to the query graph was computed and the rank of the graph belonging to the same abstract Chinese character as the query graph was determined—a rank of 1 means that the query glyph has correctly been identified. The results for the distance measure that performed best (the *landmark distance*) are listed in Table 3.1.

For evaluating the substitution similarity the same experiment²⁰ was carried out: Every pair of graphs under consideration was aligned according to the bounding boxes and then the substitution similarity Φ^{cc} was applied to the edges of the graphs. In more than 97% of the cases, the query glyph has correctly been identified. The detailed results are listed in Table 3.1.

	rank				
	1	2	10	20	200
landmark distance	85.3	91.5	97.2	98.1	99.2
substitution similarity	97.6	99.1	99.8	99.9	99.9 ²¹

Table 3.1. Performance of *substitution similarity* and of *landmark distance* on Chinese characters: percentage of items for which correct reference glyph had specified or better rank.

²⁰ The set provided by the authors actually consists of 4 178 characters in 6 fonts.

²¹ Value rounded down. Only a single item had a higher rank (actually 394).

MPEG 7 As a benchmark test for shape descriptors and the corresponding (dis-)similarity measures the *MPEG 7* data set has been used to test and compare the performance of a wide variety of approaches. Usually, each image is queried for in the set of all 1 400 images—including itself. In the resulting ranking the number of relevant items (images from same class) in a prefix of predefined length is determined. The effectiveness of an approach is often rated in terms of the so called *bulls-eye* performance, which is the overall percentage of relevant items that have been found within the prefixes of double the class size (prefix of length 40 in this case).

Table 3.2 shows results reported in [143] and [19] for some selected approaches: CSS, the contour based *curvature scale space* (see Section 3.3.1 page 141); CA, curve alignment via computing an edit distance as presented in [205]; ZM, the pixel based Zernike moments (see Section 3.3.1 page 139); SC, a skeleton based approach presented in [249]; ST, a hierarchical representation of the contour as presented in [81]. For a more comprehensive list see [19].

Among the approaches ranking images based on comparisons of the query image to each database image only, the best result (known to the author of the present work) was achieved with the hierarchical representation of the contour (ST). However, using also information about the (dis-)similarities between the database images themselves, in [19] even a bulls-eye percentage of 91.6 was achieved.

For evaluating the *substitution similarity*, as a preprocessing from every image the shape was extracted and simplified (for images containing more than one contour, the longest polyline was chosen). Then, for every pair (P_q, P_i) of polylines under consideration, from the class of similarity transformations (including transformations with reflections) a set $T = \{t_1, t_2, \dots\}$ of candidate transformations mapping P_q onto P_i was computed and the similarity was estimated as $\sigma_m = \max_{t \in T} \{\Phi^{cc}(t(P_q), P_i)\}$.²² Based on the derived ranking, the number of relevant items in the prefixes of length 40 was determined. Table 3.2 shows the results.

CSS	CA	ZM	SC	ST	SS
75.4	78.2	70.2	79.9	87.7	84.4

Table 3.2. Performance of *substitution similarity* (SS) and of some other approaches on the *MPEG 7* data set, measured by the bulls-eye percentage.

²² For details on the generation of candidate transformations see Section 4.3

3. Estimation of Shape-Similarity

UK Trademarks Finally, the *substitution similarity* was tested on the *UK trademarks* set. As a preprocessing, from each image the set of relevant shapes was extracted as described in Section 4.2. Then, for each of the 24 query images the set of 10745 images was queried in the following way: For every pair $(\mathcal{P}_q, \mathcal{P}_i)$ of sets of polylines under consideration, from the class of similarity transformations (including transformations with reflections) a set $T = \{t_1, t_2, \dots\}$ of candidate transformations mapping \mathcal{P}_q onto \mathcal{P}_i was computed and the similarity was estimated as $\sigma_m = \max_{t \in T} \{\Phi^{cs}(t(\mathcal{P}_q), \mathcal{P}_i)\}$.²² The derived ranking was rated with respect to the relevant items (the ground truth list). Table 3.3 shows the results achieved by the substitution similarity, as well as by the *ARTISAN* retrieval system [72, 74] (the individual results for each query image are listed in Table A.1 in the Appendix).

	R_n	P_n	L_n
ARTISAN	0.94	0.70	0.72
substitution similarity	0.95	0.75	0.74

Table 3.3. Performance of *substitution similarity* and of the *ARTISAN* retrieval system on *UK trademarks* set: normalized recall R_n , normalized precision P_n , and normalized last place L_n averaged over the 24 queries.

Conclusion Various experiments have shown that the *substitution similarity*—in combination with the proposed algorithm for mapping—is applicable to a broad range of problems: it outperforms other approaches in the retrieval of Chinese characters and it yields reasonable results on the *MPEG 7* data set although it is *not* specifically dedicated and restricted to the comparison of single shapes. Moreover it yields results that are even better than the ones achieved by the *ARTISAN* retrieval system on trademark images.

However, apart from the good results achieved, there are some cases—especially among the trademark images—that are problematic for this approach:

frames If the important part of a trademark image is surrounded by some kind of a simple frame, most humans do not pay much attention to that frame. The similarity measure however is influenced by it, because the frames naturally are larger than the part contained in it.

spatially independent parts Comparing two images that consist of two or more spatially independent parts and the corresponding parts are similar but arranged in slightly different ways, most humans do not observe the differences. However, it may be that there is no affine map aligning all parts properly at the same time.

3.4. Evaluation

In Section 4.4 a framework that is specifically dedicated to the retrieval of trademark images is presented. It overcomes these limitations by partitioning the images and applying the *substitution similarity* to the individual parts then.

A Framework for Automated Trademark Image Retrieval

In this chapter a framework for content-based image retrieval is presented. The framework incorporates the algorithms presented in Chapters 2 and 3, as well as an approach specifically dedicated to the comparison of trademark images.

Due to the the challenges described in Section 1.2.5 on the one hand, and the demands on trademark image retrieval described in Section 1.4.3 on the other hand, commercially offering services such as trademark search and trademark watch that rely on fully automated retrieval systems only, seems far out of sight—there is always the risk that some images cannot be recognized properly. However, automated systems may of course support human trademark examiners, for example by identifying and processing the easy cases.

The framework described in the following can be used in two ways: First, images which cannot reliably be identified as easy cases may be sorted out and handed over to manual inspection, which is recommendable in business applications. Second, all images may be processed automatically, which is the usual modus operandi in research in order to be able to compare the performance of different approaches.

4.1. Related Work

The growing need for solutions to content-based image retrieval has induced extensive research. *QBIC* (Query By Image Content), one of the first systems reaching broad publicity, was presented in 1993 (cf. [171], see also [78, 86]). Not even ten years later in [232] already 58 systems were listed. However, most of these systems focus on photographic images rather than figurative images and do not meet the demands of trademark image retrieval.

Approaches for Content-Based Trademark Image Retrieval There is a huge number of approaches that have been proposed for trademark image retrieval. For many of them it is easy to construct examples where they fail to conform to perceived similarity:

On the one hand, approaches considering individual shapes may fail when perceptually irrelevant differences in the topology lead to combinatorially relevant differences in the extraction of shapes. On the other hand, approaches considering every image as a whole may fail when frames are added or the background color is changed. On the one hand, approaches that do not consider changes in position, scale or orientation may fail when position, scale or orientation do change. On the other hand, approaches that rely on normalizing the images with respect to position, scale or orientation may fail if this normalization is affected by noise, frames or even perceptually irrelevant changes of the shapes. Approaches based on pixel intensities like using moments or histograms reflecting the spatial distribution of black and white pixels, e. g., may fail when images differ in the way of depicting the shapes (region vs. outline).

In order to reduce the impact of single failures, different approaches are often combined. Since there are no theoretical proofs for the applicability to trademark image retrieval, the effectiveness of the (combined) approaches has to be evaluated in experiments. However, there is no set of trademark images plus ground truth (query images and information about relevant images) that has been widely accepted and used for the experimental evaluation of the approaches. Therefore, an objective comparison of the effectiveness is not possible (see also [73] for a discussion). In addition, insufficient documentation of the used datasets often make an assessment of the published results impossible.

In the following, some publications dealing with content-based trademark image retrieval are shortly presented in almost chronological order.

In [57] the contours of the shapes in each image are represented by strings over a finite alphabet and images are compared using string matching techniques.

Experiments on trademark images have been carried out, but no representative results have been published.

In [245] a *System for Trademark Archival and Retrieval* named *STAR* was presented. Within this system, the shapes depicted in the images are semi-automatically extracted and each image is represented by Fourier descriptors, moment invariants, and gray level projections of the depicted shapes. Apart from annotation based retrieval, images are compared based on the individual shapes. Experiments on a set of 3000 and on a set of 500 trademark images have been carried out, but no representative results have been published.

In [227] and [127] each image is represented by an edge direction histogram, by moment invariants, and by the contours extracted from the image. After pruning based on the comparison of the edge direction histograms and the moment invariants, two images are compared according to some kind of transformational model, namely by essentially determining the costs (energy) needed to deform the first image's edges such that they match the second image's edges. The published experimental results on a set of 1100 trademark images hardly allow to assess the effectiveness of the approach, because neither the ground truth sets nor comprehensive performance indicators for the rankings have been published.

In [181] from each image the essential closed contours are extracted based on the boundaries between black and white pixels, and these contours are represented by strings over a finite alphabet coding the angles between consecutive edges. Two images are compared based on similarity values for the individual boundary strings. The published experimental results on a set of 250 trademark images hardly allow to assess the effectiveness of the approach, because they are essentially based on retrieving artificial variations of an original trademark, but no additional ground truth was used.

In [133] and [134] for the comparison, each image is represented by Zernike and pseudo Zernike moments based on the pixel intensities. Experiments that have been carried out on a set of 3000 trademark images give the impression that the approach only retrieves images with a similar global appearance, however, not distinguishing between frames and content.

In [191] apart from annotation based retrieval, images are compared using histograms based essentially on the derivatives of pixel intensities. Although experiments were carried out on a set of 63718 trademark images, the published results hardly allow to assess the effectiveness of the approach, because apparently, no ground truth sets were available.

In [11] based on the edges detected in an image, shapes are extracted. Two images are compared based on the shapes using neural networks. The published

4. A Framework for Automated Trademark Image Retrieval

experimental results on a set of 1 000 trademark images do not allow to assess the effectiveness of the approach, because no images other than the query images have been published.

In [52] and [53] for comparison, each image is represented by an edge direction histogram, by moment invariants and by information derived from a wavelet transformation. Comparison results are adapted by a relevance feedback mechanism. Experiments have been carried out using the same 1 100 trademark images as in [127]. However, the published results hardly allow to assess the effectiveness of the approach, because the ground truth sets were compiled by the authors and have not been published at all.

In [47] after normalizing with respect to translation and rotation, two images are compared based on a two-dimensional pseudo hidden Markov model. The published experimental results on a set of 401 trademark images hardly allow to assess the effectiveness of the approach, because they are essentially based on retrieving artificial variations of original trademarks, but no additional ground truth was used.

In [207] the shapes depicted in the images are semiautomatically extracted and each image is represented by feature vectors—one for each shape—containing moments, information essentially about the distances of the boundary points to the center, and an edge direction histogram. Based on these feature vectors single shapes instead of images are compared. In experiments the approach was used to retrieve deformed versions of shapes extracted from trademarks, and was tested on the *MPEG-7 core experiment CE-Shape-1* set (including the *part B* set used in the present work). However, the published results do not allow to assess the applicability of the approach to trademark image retrieval.

In [250] for the comparison, after normalizing with respect to translation and scale, each image is represented by a vector of statistical measures plus representations of the contours by strings over a fixed alphabet. The published experimental results on a set of 1 000 trademark images hardly allow to assess the effectiveness of the approach, because apparently, the ground truth sets were not compiled properly and have not been published.

In [149] from the shapes (connected black regions) in the images either their outlines or their skeletons are extracted. For each shape a degree of being one of the types *straight line segment*, *circle*, *polygon*, or *undesigned* is determined. Two images are compared based on shape-type-depending similarity values of the individual shapes and on the spatial layout. The published experimental results on a set of close to 2 000 trademark images hardly allow to assess the effectiveness of the approach, because apparently, they are essentially based on retrieving artificial variations of original trademarks and no ground truth sets have been published.

In [24] from each image a finite set of contour points (possibly with uniform spacing) is sampled and for each such point a histogram reflecting the spatial distribution of the other points is determined. For comparing two images, a transformation mapping one image onto the other is computed based on matching the histograms. The similarity of two aligned images is then estimated based on the histograms, based on the differences in brightness between matched points, or based on properties of the mapping transformation. The published experimental results on a set of 300 trademark images hardly allow to assess the effectiveness of the approach, because apparently, no ground truth sets were available.

In [153] for the comparison, after normalizing with respect to translation, rotation, and scale, each image is represented by a two-dimensional histogram reflecting essentially the spatial distribution of black or white pixels in polar coordinates. The published experimental results on a set of more than 1 000 trademark images hardly allow to assess the effectiveness of the approach, because apparently, they are essentially based on retrieving artificial variations of original trademarks and no ground truth sets have been published.

In [130] each image is represented by Zernike moments based on the pixel intensities and by a set of geometric primitives (circles, rectangles, triangles, circular arcs, straight line segments) extracted from the edges detected in the image. Two images are compared based on the moments, based on edge direction histograms, and based on the individual primitives. The approach has been tested on a set of 3 000 trademark images with manually compiled queries and has been compared with the approach presented in [24]. However, the published results hardly allow to assess the effectiveness of the approach, because apparently, the ground truth sets were compiled by the authors and have not been published at all.

In [182] the content of each image is represented by moment invariants and two histograms, one reflecting pixel intensities and the other reflecting essentially the spatial distribution of inhomogeneity. Two images are compared based on these representations of the content as well as based on additional textual information. Comparison results are adapted by relevance feedback mechanisms. Experiments focussing on the performance of different relevance feedback mechanisms have been carried out on a set of more than 250 000 trademark images and logos collected from the web. However, with respect to content-based image retrieval no representative results have been published.

In [41] for the comparison, each image is represented by size functions computed based on (binary) pixel intensities. The approach has been tested on the same dataset (*UK trademarks set*) that was used in the present work. The published results, however, are not competitive.

4. A Framework for Automated Trademark Image Retrieval

In [252] for the comparison, after normalization with respect to translation, rotation and scaling, each image is represented by moment invariants and by a histogram reflecting the spatial distribution of black and white pixels. The published experimental results on a set of 1 000 trademark images hardly allow to assess the effectiveness of the approach, because apparently they are essentially based on retrieving artificial variations of original trademarks and no ground truth sets have been published.

In [49] for the comparison, after normalizing with respect to translation (and supposedly to rotation), each image is represented by a histogram essentially reflecting the spatial distribution of black and white pixels. The published experimental results on a set of more than 2 000 trademark images hardly allow to assess the effectiveness of the approach, because apparently, no ground truth sets were available.

In [152] for the comparison, after normalization with respect to translation and scale, each image is subdivided into concentric annuli each of which is represented by Zernike moments. The published experimental results on a set of 1 000 trademark images hardly allow to assess the effectiveness of the approach, because no information about the ground truth is given.

In [114] for the comparison, after normalizing with respect to translation, scale, and rotation, each image is represented by a histogram reflecting the spatial distribution of black and white pixels, and a histogram based on angles in a triangulation of boundary corner points. The approach was tested on the *MPEG 7* data set (which was also used in the present work), and on a manually compiled set of trademark images. However, the published results on the set of trademark images hardly allow to assess the effectiveness of the approach, because apparently, the ground truth sets were not compiled properly and have not been published.

In [129] for the comparison, each image is represented by wavelet based features that essentially capture information about edge directions. The published experimental results on a set of manually collected trademark images hardly allow to assess the effectiveness of the approach, because apparently, no ground truth sets were available.

In [132] for the comparison, after normalizing with respect to translation, scale, and rotation, each image is represented by edge direction histograms and by Zernike moments. Apparently, the approach was not tested on any set of trademark images at all.

In [234] for comparison each image is represented by moment invariants and by a color histogram. Experiments were carried out on some (apparently bi-level black-and-white) trademark images, but no representative results were published.

In [238] the images are normalized with respect to translation and scale, and the contours are extracted. For the comparison, each image is represented by Zernike moments, by the variance of curvature, and by a histogram on the boundary to centroid distances. The published experimental results on a set of 1 003 trademark images hardly allow to assess the effectiveness of the approach, because apparently, the ground truth sets were compiled by the authors and have not been published.

In [186] the contours in the images are extracted and for the comparison each image is represented by a histogram on the boundary to centroid distances, and a histogram based on circumcircles in a triangulation of boundary points. Experiments have been carried out on the *MPEG 7* data set, but apparently not on trademark images.

The ARTISAN Project *ARTISAN* (Automatic Retrieval of Trademark Images by Shape ANalysis) was a project with the goal to develop and evaluate a system for automated trademark image retrieval [71] (see also [72, 73, 74]). The *ARTISAN* system is “regarded as one of the most comprehensive trademark retrieval system in the current literature” [130] because of the sophisticated extraction of perceptually relevant shapes and the elaborate evaluation of the effectiveness of different approaches to measure similarity.

From the boundaries detected in the images (at different levels of blurring), perceptually relevant image elements are extracted by grouping the boundaries according to rules based on Gestalt psychology. Two images can be compared based on shape features for the entire image as a whole, for each image element, and for each individual boundary. The shape features considered include statistical measures, Fourier descriptors, moment invariants, angular radial transform (ART) coefficients, and a curvature scale space representation. In addition, several distance measures can be chosen.

The effectiveness of the different approaches has been tested on the same dataset—the *UK trademarks* set (see Section 1.6.2.2)—that was used in the present work. For a comparison of the performances of the *ARTISAN* retrieval system and the retrieval system proposed in the present work see Table 4.1 on page 169.

4.2. Extraction of Shapes

Within this framework, the extraction of the perceptually relevant shapes in figurative images is organized as a pipeline (see Figure 4.1 for an illustration): images are vectorized by discretizing the colors and detecting boundaries of the resulting regions. Then textured regions are identified and broken lines are reconstructed. Then noise is deleted, polylines are simplified and redundant shapes are deleted.

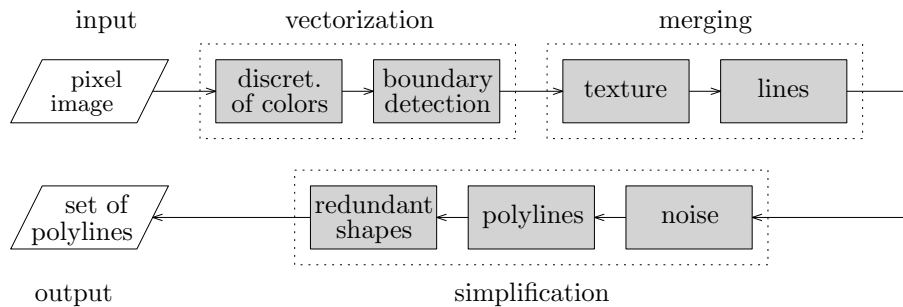


Figure 4.1. Segmentation pipeline

In order to make the process of extracting the shapes in an image independent from the resolution of the actual pixel image, respective values are computed relative to the size of the image. For trademark images the smaller of the two side lengths seems to be more relevant. However, for images with extremely large (or extremely small) aspect ratio this might give unwanted results. Therefore, in the following the size of an image I is defined as $s(I) := \max(\min(w, h), 0.25 \cdot \max(w, h))$ with w being the width and h being the height of the image in pixels.

4.2.1. Vectorization

The vectorization is carried out as described in Section 2.2. A set of suitable values for the parameters described there is given in the Appendix (Section B.1). An image classified as bi-level black-and-white based on its color histogram is segmented by thresholding. All other images run through the extensive segmentation.

In order to be classified as reliably being correct, a segmented image has to fulfill two properties: First, for any pixel the difference between its original color and its newly assigned color has to be small. Second, any edge between regions of different colors in the segmented image has to be near a large difference

of original colors. Therefore, with a *Sobel operator* (see [69, pp 271–272]) edges are detected based on the gradients,¹ and the consistency with the edges resulting from the actual segmentation is checked.² If both conditions are fulfilled—the discrepancy between colors is small and edges from different approaches do correspond—then the image is classified as reliably being correctly segmented.

4.2.2. Merging

Texture Regions of high complexity which may result from texture as well as from detailed drawings are detected as described in Section 2.3.1. A set of suitable values for the parameters described there is given in the Appendix (Section B.1). If regions of high complexity have been detected, the image is classified as possibly not correctly represented.

However, in order to allow for a completely automated extraction of shapes also, the essentially perceived shapes inside the regions of high complexity should be extracted as well. This is tried using the same approach as for the detection—with different parameters (see Section B.1) that allow for an analysis more sensitive to details—applied only to the regions detected in the first stage. Let \mathcal{R}_t be the set of pixels classified as belonging to a textured region in the second stage. Every originally detected shape with most of its boundary (the threshold is actually set to 90%) covered by \mathcal{R}_t , is deleted. In return, the shapes extracted from \mathcal{R}_t are added.

Broken Lines As a preparation for the detection of line shapes, for every shape in the image its max L_∞ -skeleton (as defined in Section 2.3.2) is computed. Let \mathcal{S}_s be the set of thin shapes—meaning that the maximum radius of an L_∞ -ball is smaller than a threshold which is actually set to $1/40 \cdot s(I)$. Let furthermore P_s be the set of skeleton points of these shapes. From this set of points a set \mathcal{S}_l of line shapes is constructed as described in Section 2.3.2. A set of suitable values for the parameters described there is given in the Appendix (Section B.1). Every originally detected shape with most of its skeleton points contained in a line shape of \mathcal{S}_l , is deleted. In return, the shapes in \mathcal{S}_l are added.

¹ For each of the three dimensions of the RGB color space, edges are detected independently and then are accumulated.

² Please note that edge detection with the Sobel operator is not assumed to be reliable in any case, but that it is assumed to confirm the other method for the easy cases.

4.2.3. Simplification

Deletion of Noise Small isolated shapes are identified and deleted as described in Section 2.4.2. The threshold θ_s on the diameter of a shape is set to the maximum of $s(I)/100$ and $d_{75}/1.5$ with d_{75} being the diameter of the 75th-biggest shape in the Image. The threshold $\theta_d(S_1, S_2)$ on the distance between two shapes S_1 and S_2 is set to $3 \cdot \min(\text{diam}(S_1), \text{diam}(S_2))$ with $\text{diam}(\times)$ being the diameter. The threshold θ_c on the diameter of a cluster is set to $s(I)/40$.

Polyline Simplification The polylines representing the shapes are simplified using the *adaptive coarsening plus corner simplification* as described in Section 2.4.1 page 103. A set of suitable values for the parameters used within the algorithm is given in the Appendix (Section B.1).

Given a polyline P in an image I , the error threshold Δ_{max} for the polyline simplification is basically chosen depending on the image size $s(I)$. However, to prevent distorting small shapes and small features too much, a factor $f_s \leq 1$ considering the relative size, and a factor $f_l \leq 1$ considering the expected richness of detail is introduced. Let $d(P)$ be the maximum extent of P in x or in y direction, then $f_s := (1 - d(P)/s(I)) \cdot 0.2 + d(P)/s(I) \cdot 1$ restricted to the range $[0, 1]$. Let furthermore $l(P)$ be the length of P , then $f_l := 4 \cdot d(P)/l(P)$ restricted to the range $[0.25, 1]$. Finally, the error threshold Δ_{max} is determined as the minimum of $0.01 \cdot s(I) \cdot f_s \cdot f_l$ and $0.25 \cdot d(P)$. With these values, for instance the depiction of a circle is represented by a polygon with 18 edges if its diameter is 1/2 times the image size, 14 edges if its diameter is 1/8 times the image size, and 8 edges if its diameter is 1/32 times the image size.

Deletion of Redundant Shapes Shapes subordinate to bigger shapes are identified and deleted as described in Section 2.4.2. The threshold θ_n on the distance of the boundary of a—potentially—subordinate shape S to the boundary of the shape R that it might be subordinate to, depends on the the size of the shape R . It is set to be the minimum of $s(I)/50$ and $d(R)/20$ with $d(R)$ being the maximum extent of R in x or in y direction.

4.3. Similarity Estimation based on Mapping

In order to build a general purpose retrieval system for figurative images, the polyline sets obtained from the extraction pipeline described in Section 4.2 are directly used as input for the estimation of similarity based on mapping

4.3. Similarity Estimation based on Mapping

and the *substitution similarity* according to Sections 3.2.2 and 3.3.3. A set of suitable values for the parameters described there is given in the Appendix (Section B.2).

Given two sets \mathcal{P}_1 and \mathcal{P}_2 of polylines, from the class of similarity transformations (including transformations with reflections) a set $T = \{t_1, t_2, \dots\}$ of candidate transformations mapping \mathcal{P}_1 onto \mathcal{P}_2 is computed and the similarity is estimated as $\sigma_m = \max_{t \in T} \{\Phi^{cs}(t(\mathcal{P}_1), \mathcal{P}_2)\}$.

For determining similarity transformations, the number of votes is set to $250 \cdot \max(n_1, n_2)$ with $n_1 = |V(\mathcal{P}_1)|$ and $n_2 = |V(\mathcal{P}_2)|$ being the numbers of vertices of the two shape sets. Although the number of vertex pairs is $n_1 \cdot n_2$, informally speaking, for a pair of polylines (P_i, P_j) from $\mathcal{P}_1 \times \mathcal{P}_2$ that is sufficiently similar, there are $\min(|P_i|, |P_j|)$ pairs that may lead to the same augmented sample. In the experiments performed on the *MPEG 7* data set and on the *UK trademarks* set the chosen number of votes proved to be high enough to yield reasonably good results (see Section 3.4).

In order to obviate the evaluation of spurious transformations, the number of candidates in T has to be bounded: For ω_{max} being the maximum cluster weight, only clusters with weight greater than $0.1 \cdot \omega_{max}$ and rank smaller or equal to 50 are considered. Of course, there is no guarantee that no promising transformation gets excluded by that. However, on average for relevant images³ the best transformation is among the first 50 clusters in 72% of the cases, and the increase of the similarity value achieved by considering more than 50 clusters is below 5% in 92% of the cases. For relevant images that are in fact similar with respect to the *substitution similarity* (maximum value of at least 0.8) the best transformation is among the first 50 clusters even in 86% of the cases, and the possible increase is below 5% in 98% of the cases. This also indicates that the algorithm for finding candidate transformations and the *substitution similarity* do fairly well harmonize.

Although the definition of the *substitution similarity* is not specifically dedicated to the retrieval of trademark images, the results achieved in experiments on the *UK trademarks* set (see Section 1.6.2.2) are better than the ones achieved by the *ARTISAN* retrieval system [72, 74] (see Table 4.1 on page 169 for the average values and Table A.1 on page 175 for the individual results for each query image).

³ For the *UK trademarks* set every query image was compared to the respective relevant images 50 times. For the *MPEG 7* data set every image was compared to the 20 images from the same class. In both cases the first 500 candidate transformations were evaluated.

4.4. Similarity Evaluation based on Image Primitives

In order to build a retrieval system specialized on trademark images, further domain knowledge on figurative images, particularly on trademark images, is exploited for the estimation of similarity:

- Trademark images may be perceived as being virtually the same when they contain the same basic shapes, just with different gaps inbetween. Moreover, trademark images may also be perceived to be similar because they contain the same basic shapes, even when the spatial arrangement and/or the relative sizes of the shapes are different.
- Trademark images may be perceived as being virtually the same, regardless of the existence or the shape of frames (see Section 1.2.3, page 28).
- Trademark images may be perceived to be similar because they contain the same arrangements of shapes, even when the shapes are different (see [148]).
- In trademark images, basic geometric figures such as triangles, rectangles, circles, etc. are identified and may lead to the perception of similarity even if they are slightly distorted, incomplete, or composed of several parts.

Figure 4.2 shows examples of trademark images illustrating some of the statements above: the images in the top row are query images from the *UK trademarks* set (see Section 1.6.2.2), the images in the bottom row are relevant (similar) images with (a) a different arrangement, (b) an additional frame, (c) composed figures, (d) a different number of copies, and (e) different relative sizes.

4.4.1. Idea of the Proposed Approach

The idea is to decompose the trademark images into their basic perceptual units (as has also been recommended by the *ARTISAN* project [74]), to identify frequently used geometric figures, and to determine the relationships between the units within each image. Images may then be compared based on such a representation by comparing individual units and corresponding relationships and to derive an overall similarity value from the individual results. However, since the extraction of high level features is generally not robust and may lead to different representations of similar images, the estimation of similarity based on the decompositions is backed by the similarity estimation based on mapping

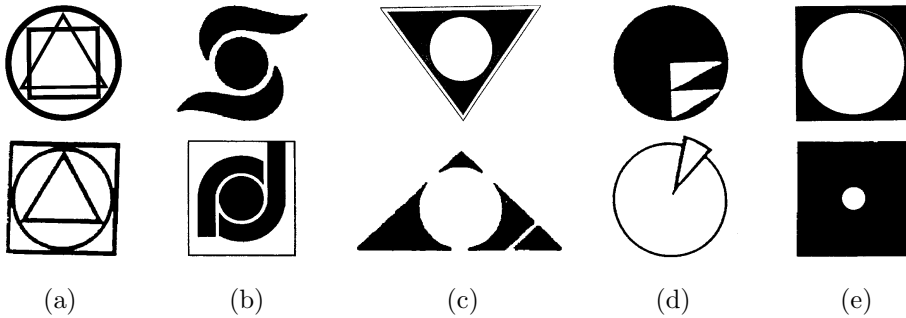


Figure 4.2. Similar trademarks that pose a challenge to straightforward application of standard (dis-)similarity measures.

as described in Section 4.3. In the following, the main ideas of the approach are described, while minutiae of the implementation are given in the Appendix (Section B.3).

4.4.2. Basic Perceptual Units and Relationships

According to the *recognition-by-components* theory, humans recognize three-dimensional objects based on partitioning them into simple components and identifying basic primitives, called *geons*, plus their relationships [28]. There is also a huge literature on describing two-dimensional scenes based on image primitives and their relationships (see, e.g., [177] and [90]). Using a finite alphabet of symbols describing primitives and relationships, a shape or an image is then described by a string over this alphabet. In order to get small alphabets that are still capable of describing arbitrary shapes the primitives are usually chosen to be very simple, e.g., straight line segments or circular arcs. Forming representations of shapes that are suitable for similarity estimation from these simple primitives, however, is very challenging (see [177] for a discussion).

As opposed to approaches using such restricted alphabets, the target pursued here is to allow arbitrary units that are perceptually relevant. These units might be spatially independent shapes (or sets of shapes), as well as salient geometric figures. Analyzing a collection of 1 762 395 trademark images showed that the most frequently occurring two-dimensional figures are rectangles (in 23 % of the images), circles (15 %), irregular quadrilaterals (12 %), ellipses (9 %), squares (8 %), and triangles (≥ 4 %).⁴

⁴ Counted based on the assigned Vienna codes. Results were kindly provided by Aktor Knowledge Technology.

4. A Framework for Automated Trademark Image Retrieval

In the following an instantiation of a primitive will be called *figure*. The image primitives considered here are:

- rectangles (as a generalization of squares)
- ellipses (as a generalization of circles)
- triangles
- arbitrary convex polygons
- arbitrary sets of polylines

When a shape is depicted by its outline, the extraction of shapes from the image may result in two equidistant polylines rather than a single one. Since line thickness is subject to the designing of figurative images, the distance between these two polylines may take an almost arbitrary value. Therefore— analogously to concentric circles—‘concentric’ ellipses, rectangles, triangles, and convex polygons respectively, are conflated to a single figure with multiple layers.

For a pair (F_i, F_j) of figures the relationship $R_{i,j}$ stores information about

- the size of F_j relative to F_i
- the relative distance between F_i and F_j
- whether F_i and F_j are similar (with respect to *substitution similarity*) under translations, rotations (actually under rigid motions) and under reflections, respectively.

4.4.3. Extraction of Figures

Given a set \mathcal{P} of polylines representing the shapes in a figurative image, a set \mathcal{F} of figures is extracted as described in the following. Based on the minimum distance between points on the polylines, and a threshold on this distance, the polylines from \mathcal{P} are grouped according to a single linkage clustering.⁵ Let \mathcal{C} be the cluster with largest perimeter of the convex hull of its elements. Four different options are considered:

1. Introducing a figure representing the convex hull of the polylines in \mathcal{C} . Possible types of figures are rectangle, ellipse, triangle, and convex polygon.

⁵ Let $G = (\mathcal{P}, E)$ be the graph that has the set of polylines under consideration as vertex set, and an edge for each two polylines with minimum distance smaller than the threshold, or formally $E = \{\{P_1, P_2\} \mid \exists p_1 \in P_1, p_2 \in P_2 \text{ such that } \|p_1 - p_2\| < \theta_d\}$ for θ_d being the threshold on the distance. The single linkage clusters correspond to the connected components in G .

4.4. Similarity Evaluation based on Image Primitives

2. Introducing a figure for a perceptually relevant geometric primitive that has been detected at an arbitrary position. Possible types of figures are square, circle, or equilateral triangle.
3. Introducing an individual figure for each polyline in \mathcal{C} . Possible types of figures are rectangle, ellipse, triangle, convex polygon, and arbitrary polyline.
4. Introducing a single figure representing the whole cluster.

The option that is supposed to lead to the most appropriate decomposition of the image is chosen and the respective figures are added to \mathcal{F} . The polylines belonging to \mathcal{C} are removed from \mathcal{P} , and in the first two cases, the parts of the polylines belonging to \mathcal{C} but not properly represented by the figure are added to \mathcal{P} again. This process is repeated until \mathcal{P} is empty. Finally, for every pair $(F_i, F_j) \in \mathcal{F} \times \mathcal{F}$ with $i \neq j$ a relationship $R_{i,j}$ is created. Figure 4.3 shows some examples of trademark images and their representations by image primitives.

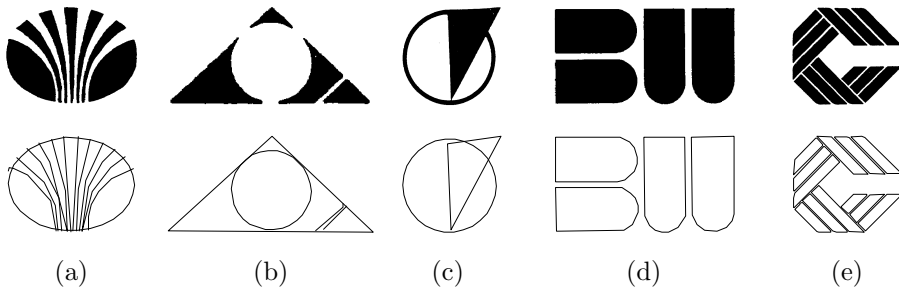


Figure 4.3. Representation by image primitives:
 (a), (b) images for which introducing a figure representing the convex hull was rated best in the first step, (c) image for which introducing a figure for a geometric primitive at an arbitrary position was rated best in the first step, (d) image for which the representation by 4 individual figures was rated best, (e) image for which the representation by a single set of shapes was rated best.

The threshold on the distance in the clustering, as well as the valuation of the different options are adapted in such a way that the number of figures in \mathcal{F} does not exceed a certain threshold—in the current implementation this threshold is set to 8. On the one hand, an arbitrary large number of figures would lead

4. A Framework for Automated Trademark Image Retrieval

to unfeasible effort during the comparison of images (see Section 4.4.4), on the other hand, in a figurative image the number of distinct entities that are perceptually relevant may not be arbitrarily large.

Every figure and every relationship gets a weight that shall reflect the perceptual importance. The weight $\omega(F)$ of a figure F depends on its size (the greater diameter and length, the more important), on whether it might be a frame (frames are less important), and on whether it is a translated copy of another figure (copies are less important). The weight $\omega(R)$ of a relationship R depends on the weight of the two figures it relates and on the relative distance between the two figures (the farther apart two figures, the less important the relationship between them).

4.4.4. Comparison of Images

Let $I_1 = (\mathcal{F}_1, \mathcal{R}_1)$ and $I_2 = (\mathcal{F}_2, \mathcal{R}_2)$ be two image representations with n_1 and n_2 figures, respectively. Furthermore, without loss of generality, let $n_1 \leq n_2$ and for both images let $\omega_{\mathcal{F}} := \sum_{F \in \mathcal{F}} \omega(F)$ and $\omega_{\mathcal{R}} := \sum_{R \in \mathcal{R}} \omega(R)$ such that $\omega_{\mathcal{F}} + \omega_{\mathcal{R}} = 1$.

A matching between \mathcal{F}_1 and \mathcal{F}_2 is an injective function $m: \{1, \dots, n_1\} \rightarrow \{1, \dots, n_2\}$. It uniquely assigns every figure $F_{1,i}$ of \mathcal{F}_1 exactly one figure $F_{2,m(i)}$ of \mathcal{F}_2 and therefore, implicitly also assigns every relationship $R_{1,i,j}$ of \mathcal{R}_1 a relationship $R_{2,m(i),m(j)}$ of \mathcal{R}_2 .

Based on an underlying measure $\sigma_{\mathcal{F}}: \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$ of similarity between figures (here basically the *substitution similarity* as defined in Section 3.3.3 is used), and a measure $\sigma_{\mathcal{R}}: \mathcal{R} \times \mathcal{R} \rightarrow \mathbb{R}$ of similarity between relationships, the similarity of I_1 and I_2 is then defined as the maximum sum of individual weighted similarities over all matchings:

$$\sigma_p(I_1, I_2) := \max_{m: \text{matching}} \left\{ \sum_{1 \leq i \leq n_1} \sigma_{\mathcal{F}}(F_{1,i}, F_{2,m(i)}) \cdot \frac{\omega(F_{1,i}) + \omega(F_{2,m(i)})}{2} + \sum_{1 \leq i \leq n_1} \sum_{\substack{1 \leq j \leq n_1 \\ j \neq i}} \sigma_{\mathcal{R}}(R_{1,i,j}, R_{2,m(i),m(j)}) \cdot \frac{\omega(R_{1,i,j}) + \omega(R_{2,m(i),m(j)})}{2} \right\}$$

For arbitrary measures $\sigma_{\mathcal{F}}$ and $\sigma_{\mathcal{R}}$ of similarity, the problem of deciding whether the best matching yields a value that is greater or equal a given threshold is an extension of the so called *quadratic assignment problem*⁶ which is known to be NP-complete (see, e. g., [144]). However, due to the bound on the

⁶ Given two $n \times n$ matrices A, B over \mathbb{R}_0^+ , find an $n \times n$ permutation matrix X such that $\sum_{i,j} A[i][j] \cdot (XB X^T)[i][j]$ is minimized.

number of figures in an image representation, the asymptotic time complexity of algorithms computing the best matching is not an issue. The optimal value can simply be determined by enumerating and evaluating all possible matchings.

4.4.5. Experimental Results

The similarity estimation based on image primitives was tested on the *UK trademarks* set (see Section 1.6.2.2). As a preprocessing, from each image I the set \mathcal{P} of relevant polylines was extracted as described in Section 4.2 and from these polylines, the sets \mathcal{F} and \mathcal{R} of figures and relationships were determined.

Then, each of the 24 query images was queried for in the set of 10 745 images in the following way: First, for every pair (I_q, I_i) of images under consideration, using \mathcal{P}_q and \mathcal{P}_i the *substitution similarity* σ_m based on mapping was computed according to Section 4.3. Second, for every pair (I_q, I_i) of images under consideration, using $\mathcal{F}_q, \mathcal{F}_i, \mathcal{R}_q,$ and \mathcal{R}_i the primitive based similarity σ_p was computed. Both values were combined following an idea derived from the *dynamic partial function* (see Section 1.3.4 page 47). A high value of either one measure of similarity gives evidence that the two images are perceived to be similar. A low value, on the other hand, may correspond to low perceptual similarity or to a weakness of the similarity measure. Therefore, a weighted sum is computed where the weight ω depends on the similarity value itself: $\sigma_c := (\sigma_m \cdot \omega_m + \sigma_p \cdot \omega_p) / (\omega_m + \omega_p)$, with $\omega_m := \sigma_m$ and $\omega_p := \sigma_p$. The derived ranking was rated with respect to the relevant items (the ground truth list). Table 4.1 shows the results achieved by the *substitution similarity* based on mapping, by the combined approach, and by the *ARTISAN* retrieval system [72, 74] (the individual results for each query image are listed in Tables A.1 and A.2 in the Appendix).

	R_n	P_n	L_n
ARTISAN	0.94	0.70	0.72
substitution similarity	0.95	0.75	0.74
combined approach	0.96	0.78	0.80

Table 4.1. Performance of the *substitution similarity* σ_m based on mapping, of the combined approach σ_c , and of the *ARTISAN* retrieval system on the *UK trademarks* set: normalized recall R_n , normalized precision P_n , and normalized last place L_n averaged over the 24 queries.

4.5. Conclusion and Future Work

A comprehensive framework for content-based trademark image retrieval has been developed. The extraction of shapes copes with the main challenges of real-world images, namely with colors, blurred edges, compression artefacts, texture, broken lines and noise. The proposed combination of similarity estimation based on image primitives and the *substitution similarity* shows a high conformance with perceived similarity and it facilitates significantly better retrieval results than previous approaches.

Possible directions for future work include the development of an additional pruning stage like suggested in [227]. Entirely unsimilar images might be sorted out based on a suitable set of simple shape descriptors, such that the highly discerning but more expensive similarity measure only has to be applied to a limited number of images.

Moreover, the set of primitives considered for the decomposition of images might be augmented by other geometric figures. If the extraction of shapes is carried out in a semiautomatic way, even arbitrary types of figurative elements listed in the Vienna Classification (see Section 1.4.3.2) could be incorporated. In this way it would be possible to fuse descriptive information and content-based information in a single system for image retrieval, instead of using both types of information separately.

Experiments on the UK Trademarks Set

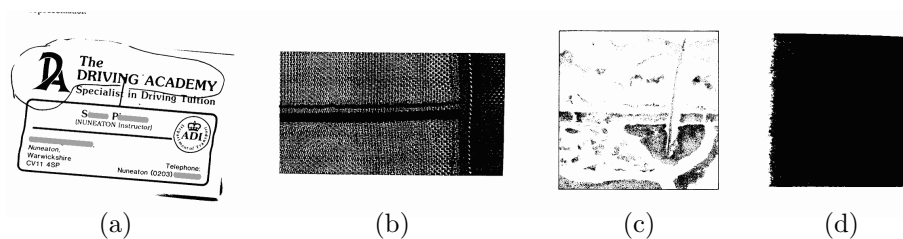


Figure A.1. Strange images:
Images where (a) not only a trademark is depicted, (b) a texture pattern rather than a figurative image is depicted, (c) it is not clear what is depicted, and (d) it is not clear if something is depicted at all.

A. Experiments on the UK Trademarks Set

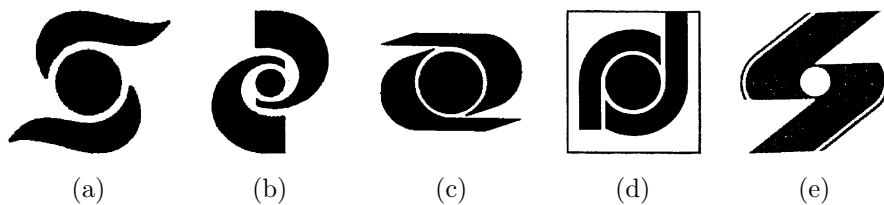


Figure A.2. Strange reference query I:
(a) query image, (b), (c) images not contained in list of relevant images, (d), (e) images contained in list of relevant images.

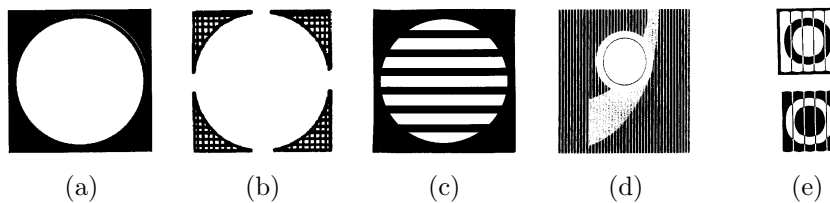


Figure A.3. Strange reference query II:
(a) query image, (b), (c) images not contained in list of relevant images, (d), (e) images contained in list of relevant images.

A. Experiments on the UK Trademarks Set

























	query	σ_m			σ_c		
		R_n	P_n	L_n	R_n	P_n	L_n
1	 1037814	0.97	0.84	0.86	0.97	0.82	0.86
2	 1055261	0.94	0.86	0.41	0.98	0.89	0.69
3	 1138103	0.97	0.77	0.84	0.97	0.83	0.77
4	 1138293	0.87	0.79	0.00	0.87	0.79	0.00
5	 1190540	0.92	0.57	0.74	0.99	0.82	0.97
6	 1259886	0.97	0.91	0.49	0.97	0.88	0.57
7	 1267206	0.98	0.74	0.91	0.99	0.76	0.96
8	 1279931	0.98	0.82	0.88	0.96	0.76	0.83
9	 1289047	1.00	0.97	1.00	1.00	0.97	1.00
10	 1376861	0.93	0.65	0.78	0.95	0.64	0.78
11	 1439229	0.98	0.89	0.81	1.00	0.91	0.99
12	 1445511	1.00	1.00	1.00	1.00	1.00	1.00
13	 1486213	0.91	0.47	0.69	0.93	0.50	0.78
14	 1525429	0.97	0.75	0.89	0.98	0.71	0.93
15	 1575268	1.00	0.89	0.99	1.00	0.93	0.99
16	 2010916	0.99	0.84	0.96	1.00	0.88	0.99
17	 2016658	0.93	0.59	0.61	0.95	0.62	0.67
18	 2018809	0.97	0.56	0.93	0.99	0.87	0.96
19	 2042822	0.59	0.22	0.02	0.63	0.24	0.04
20	 3289	0.98	0.66	0.90	0.99	0.72	0.96
21	 392632	1.00	0.94	0.99	1.00	0.90	0.99
22	 665322	0.98	0.67	0.95	0.97	0.64	0.94
23	 914	0.97	0.78	0.87	0.99	0.83	0.92
24	 967049	0.90	0.82	0.33	0.95	0.85	0.50
	average	0.95	0.75	0.74	0.96	0.78	0.80

Table A.1. Performance on UK trademarks set: 24 query images plus values of normalized recall R_n , normalized precision P_n , and normalized last place ranking L_n for the substitution similarity σ_m based on mapping and for the combined approach σ_c .

A. Experiments on the UK Trademarks Set

1	rel.						
	ret.						
	2	rel.					
		ret.					
3		rel.					
		ret.					
	ret.						

Table A.2. Performance on UK trademarks set: relevant images and first 10 images retrieved using the combined approach σ_c (ignoring duplicates).

4	rel.				
	ret.				
5	rel.				
	ret.				
6	rel.				
	ret.				

Table A.2. Performance on UK trademarks set (continued): relevant images and first 10 images retrieved using the combined approach σ_c (ignoring duplicates).

A. Experiments on the UK Trademarks Set

7	rel.					
	ret.					
8	rel.					
	ret.					
9	rel.					
	ret.					

Table A.2. Performance on UK trademarks set (continued): relevant images and first 10 images retrieved using the combined approach σ_c (ignoring duplicates).

A. Experiments on the UK Trademarks Set

10	rel.					
	ret.					
11	rel.					
	ret.					
12	rel.					
	ret.					

Table A.2. Performance on UK trademarks set (continued): relevant images and first 10 images retrieved using the combined approach σ_c (ignoring duplicates).

A. Experiments on the UK Trademarks Set

13	rel.					
	ret.					
14	rel.					
	ret.					
15	rel.					
	ret.					

Table A.2. Performance on UK trademarks set (continued): relevant images and first 10 images retrieved using the combined approach σ_c (ignoring duplicates).

A. Experiments on the UK Trademarks Set

16	rel.						
	ret.						
	17	rel.					
		ret.					

Table A.2. Performance on UK trademarks set (continued): relevant images and first 10 images retrieved using the combined approach σ_c (ignoring duplicates).

A. Experiments on the UK Trademarks Set

18	rel.					
	ret.					
19	rel.					
	ret.					

Table A.2. Performance on UK trademarks set (continued): relevant images and first 10 images retrieved using the combined approach σ_c (ignoring duplicates).


































20	rel.					
	ret.					
21	rel.					
	ret.					
22	rel.					
	ret.					
						

Table A.2. Performance on UK trademarks set (continued): relevant images and first 10 images retrieved using the combined approach σ_c (ignoring duplicates).

A. Experiments on the UK Trademarks Set

23	rel.					
	ret.					
24	rel.					
	ret.					

Table A.2. Performance on UK trademarks set (continued): relevant images and first 10 images retrieved using the combined approach σ_c (ignoring duplicates).

Specification of Parameter Values

B.1. Extraction of Shapes

This section gives a set of suitable values for the parameters of the extraction of shapes described in Chapter 2.

Vectorization

For simplicity, all computations concerning color and brightness are based on an *RGB* model using the ranges $[0, 255]$. The distances between colors are computed as Euclidean distances between RGB triples. Due to the non-linearity of the perception of intensities (see Section 1.2.1) along with the calibration of display devices it seemed to be beneficial to rescale all values according to $I_{rescaled} = (I_{original}/255)^\gamma \cdot 255$ using $\gamma = 1.25$. However, all computations might also be carried out based on distances between colors in the *Lab* color space¹. Distances between pixels are also computed as Euclidean distances.

An image is classified as bi-level black-and-white image if the occurring colors can be grouped into two classes \mathcal{B} and \mathcal{W} such that the minimum difference in brightness between any element of \mathcal{B} and any element of \mathcal{W} is greater than 128 (half the maximum possible brightness), the maximum difference in brightness

¹ In *Lab* color space, as standardized by the *International Commission on Illumination (Commission internationale de l'éclairage – CIE)* in 1976, equal Euclidean distances between representations of colors almost correspond to equal perceived differences between colors (cf. [64]).

B. Specification of Parameter Values

within each color class is smaller than 10, and the maximum deviation of any color from a gray tone (which corresponds to the chroma of the color) is smaller than 20.

Given a pixel p_0 , the following notations and definitions for properties of this pixel are used in this section:

- The neighborhood $\mathbf{n}(p_0)$ is the set of (maximally four) pixels that are edge-neighboring p_0 ; the neighborhood $\mathbf{n}(C)$ of a set C of pixels is the union of the pixels' neighborhoods without C , formally $\mathbf{n}(C) := \bigcup_{p \in C} \mathbf{n}(p) \setminus C$. The r -neighborhood $\mathbf{n}_r(p_0)$ is the set of pixels with distance at most r (that means $\mathbf{n}(p_0)$ is the 1-neighborhood of p_0 without p_0 itself).
- $\mathbf{col}_o(p_0)$ is the *original color* of the pixel, whereas $\mathbf{col}_a(p_0)$ is the *discrete color* the pixel has been assigned to. The *blurred color* $\mathbf{col}_b(p_0)$ is the weighted average of the original colors of the r_b -neighborhood of p_0 for some radius r_b . Computing the blurred colors for the whole image simply corresponds to applying a low-pass filter.
- A measure for the the variation of colors within the neighborhood of a pixel can be derived from the differences of the original colors² as $\mathbf{var}(p_0) := \frac{1}{4} \sum_{p \in \mathbf{n}(p_0)} \min(\|\mathbf{col}_o(p) - \mathbf{col}_o(p_0)\|^2/t_v, 1)$ with t_v being a threshold on the maximal difference considered. For a given radius r , the *ditheredness* $\mathbf{var}_r(p_0)$ of a pixel p_0 may then be captured by the average of the variations of the pixels in the r -neighborhood of p_0 . The ditheredness $\mathbf{var}(C)$ of a set C of pixels may be captured by the average of the variations of the pixels in C .
- The value $\mathbf{sub}(p_0)$ as derived from the pixel's closeness to another pixel of very different color shall give an approximation for the degree of being subordinate to a boundary between two regions of very different colors, meaning to which extend small differences of colors are not perceived because of a much bigger contrast in the vicinity.

For determining the blurred version of an image I with image size $s(I)$ a conic kernel with radius $s(I)/400$ (1.5 for small images) is used. For determining the ditheredness of a pixel, the threshold t_v is set to $48 \cdot 48$. For determining the closeness of a pixel to a boundary between a great change in color, all pixels with distance smaller than $s(I)/75$ (4 for small images) are considered.

² The idea is similar to the concept of *image energy* as used in [113], however, only a global value for the complete image was computed there.

The 12 steps of the color discretization process are the following:

1st step: region growing Let the *maximal blurred color distance* $\text{cd}_r(p_0)$ within radius r of a pixel p_0 be defined as the maximal distance between the blurred color of any pixel in the r -neighborhood of p_0 , and the blurred color of p_0 :

$$\text{cd}_r(p_0) := \max_{p \in \mathfrak{n}_r(p_0)} \{ \|\text{col}_b(p) - \text{col}_b(p_0)\| \}$$

Pixels with small maximal blurred color distance are candidates for seeds of new clusters. All pixels are sorted according to their maximal blurred color distance with respect to a radius r_s , which is set to $s(I)/150$ (2 for small images). As long as the minimum for the hitherto unclassified pixels is sufficiently small (≤ 10), the corresponding pixel is taken as seed p_s of a new cluster $C := \{p_s\}$ and the color of that cluster—meaning the discrete color of its elements—is set to $\text{col}_b(p_s)$.

Once a cluster has been created it gets enlarged. While there is a pixel with sufficiently small maximal blurred color distance in the neighborhood of the cluster, such that the difference of the pixel’s color to the cluster’s color is sufficiently small, the pixel is added to the cluster and the neighborhood is updated. The threshold on the maximal blurred color distance for adding a pixel to a cluster is set to 10 and—depending on the seed pixel’s ditheredness—is increased by a factor between 1 and 4. The threshold on the difference in color for adding a pixel to a cluster is set to 20 and—depending on the seed pixel’s ditheredness and on the candidate pixel’s closeness to a boundary between very different colors—is increased by a factor between 1 and 8.

2nd step: enlarging and merging clusters For every unclassified (meaning not belonging to any cluster) pixel p_0 that has a sufficiently small maximal blurred color distance, and that is adjacent to a cluster C (there might be multiple such clusters), the priority for being added to C is computed. While the minimum (best) priority is sufficiently small, the corresponding pixel is added to the cluster and the priorities of its neighbors are updated.

The radius r_g for computing the maximal blurred color distance is set to $s(I)/300$ (1 for small images). The threshold on the maximal blurred color distance for adding a pixel to a cluster is set to 10 and—depending on the seed pixel’s ditheredness—is increased by a factor between 1 and 4. The priority is computed as the maximum (the inferior one) of two values. First, the difference between the cluster’s color and the pixel’s blurred color which—depending on the number of common edges, on the cluster’s ditheredness, and on the pixel’s ditheredness—gets decreased by a factor

B. Specification of Parameter Values

between 1 and $1/16$. Second, the difference between the cluster's color and the pixel's original color which—depending on the number of common edges, on the cluster's ditheredness, on the pixel's ditheredness, and on the pixel's closeness to a boundary between very different colors—gets decreased by a factor between 1 and $1/32$. The threshold on the priority for a pixel to be added to a cluster is set to 4.

After the clusters have been enlarged, neighboring clusters with sufficiently small difference in color are merged. For any two neighboring clusters, if the difference in color is smaller than a threshold which is set to 5 and—depending on the clusters' ditheredness—increased by a factor between 1 and 4, the priority for the two clusters to be merged is computed as the difference in colors which—depending on the clusters' ditheredness and on the number of common edges—gets decreased by a factor between 1 and $1/8$. While there are still pairs of clusters that may be merged, the pair with minimum priority is determined, merged, and the priorities of the other clusters are updated. The color of a resulting cluster is simply the average color of its pixels. In the end, clusters that are smaller than $1/800$ times the number of pixels in the image (smaller than 2 for small images) are deleted.

3rd step: detecting border pixels In order to distinguish between border pixels and pixels from the inside of thin shapes, for each pixel (that has not yet been assigned a color) its vicinity is searched for evidences that the pixel is an intermediate one: For a pixel p_0 with color c_0 and coordinates (x_0, y_0) , the considered pairs are $((x_0 - r, y_0), (x_0 + r, y_0))$ and $((x_0, y_0 - r), (x_0, y_0 + r))$ with $r \leq s(I)/150$ (≤ 2 for small images). Let (p_+, p_-) be one of these pairs and let c_+ and c_- be the respective colors. Based on the distance of c_0 to the line segment $\overline{c_+c_-}$ in color space, the degree of being border pixel between p_+ and p_- is determined as $b(p_0, p_+, p_-) := \max(0, \min(1, (\|c_+ - c_-\| - \text{dist}(c_0, \overline{c_+c_-})/80)))$.

Based on the evidences for all the pairs under consideration, a value $b(p_0)$ for the degree of being a border pixel is determined from the maximum $b_{max}(p_0)$ for all considered pairs (p_+, p_-) and the average $b_{avg}(p_0)$ of the maxima for each considered distance r as $b(p_0) := \sin^8(\pi/2 \cdot (0.75 \cdot b_{max} + 0.25 \cdot b_{avg}))$.

4th step: further enlarging and merging clusters The computation of the pixels' priorities and the clusters' priorities is performed as in the 2nd step. However, whether a pixel might be added to a cluster now does not depend on its maximal blurred color distance any more, but on the pixel's degree of being a border pixel: The threshold on the priority of a pixel p_0 to be added to a cluster is set to $4 \cdot (1 - b(p_0))$.

5th step: uniqueness In order to determine how unique (meaning how independent from ‘neighboring clusters’) a pixel is, the colors of the clusters that are not far away and that are not blocked by other clusters are examined. For an unclassified pixel p_0 let $N_1(p_0) := \mathbf{n}(p_0)$ be the set of pixels that can be reached by crossing one edge. Furthermore, let $N_1^u(p_0) \subseteq N_1(p_0)$ be the unclassified ones and $N_1^c(p_0) \subseteq N_1(p_0)$ be the classified ones. The set $N_k(p_0)$ of relevant pixels that can be reached by crossing at most k edges is defined recursively as $N_k(p_0) = N_{k-1}(p_0) \cup \mathbf{n}(N_{k-1}^u(p_0))$. The sequence $(N_1^c(p_0), N_2^c(p_0), \dots)$ induces a sequence of minimum color differences to p_0 that is monotone decreasing.

The maximum distance r_u of pixels considered is $s(I)/50$ (2 for small images). For distance k , let the minimum normalized color distance $cd_k(p_0)$ be $\min_{p \in N_k^c(p_0)} \{ \min(\|\text{col}_o(p_0) - \text{col}_d(p)\|/150, \|\text{col}_b(p_0) - \text{col}_d(p)\|/100) \}$ restricted to $[0, 1]$. The uniqueness of the color of a pixel p_0 is then determined as $u(p_0) = 1 - 1/r_u \cdot \sum_{k=1}^{r_u} cd_k(p_0)^4$.

6th step: merging In addition to the existing clusters, every unclassified pixel constitutes a (preliminary) cluster. For every pair of neighboring clusters the priority of being merged is computed as the distance of the clusters’ colors, which—depending on the clusters’ intern variation of colors—is increased by a factor between 1 and 2, and—depending on the sizes of the cluster—is decreased by a factor between 1 and 1/4, and—depending on the clusters’ ditheredness—is decreased by a factor between 1 and 1/4, and—depending on the number of common edges—is decreased or increase by a factor between 0.5 and 2, and—depending on the clusters’ pixels’ closeness to a boundaries between very different colors—is decreased by a factor between 1 and 1/2, and—depending on the clusters’ pixels’ uniqueness and degree of being border pixels—increased by an additive value between 0 and the maximum allowed priority (32). While the minimum priority is sufficiently small (≤ 32), the corresponding two clusters are merged and the priorities of the pairs containing one of them are updated.

In the end, all pixels belonging to clusters that are smaller than 1/8 000 times the number of pixels in the image (smaller than 2 for small images) are marked as unclassified again.

7th step: further enlarging and merging clusters The same values as in the 4th step are used.

8th step: merging clusters based on color Maximal sets of clusters such that the pairwise differences in color are sufficiently small are determined greedily, and these clusters are merged. The threshold on the distance in color is set to 20.

B. Specification of Parameter Values

9th step: handling antialiasing Let c_0 be the color of pixel p_0 and let $V := \{k \cdot \pi/4 \mid k = 1, \dots, 8\}$ be the (angles of the) directions under consideration. Starting from p_0 , every direction is searched for the nearest pixel belonging to a cluster. Let p_+ and p_- be two such pixels located in opposite directions from p_0 , and let c_+ and c_- be the colors of the corresponding clusters. If both, the distance of c_0 from the line segment $\overline{c_+c_-}$ in color space, and the distance $\|p_+ - p_-\|$ are sufficiently small, this gives an indication that p_0 is a border pixel between colors c_+ and c_- . The threshold θ_d on the distance for determining whether a pixel p_0 with color c_0 has an intermediate color is set to $s(I)/75$ (2 for small images).

As a consequence of the fact that in many images, the colors have reduced chroma³ near boundaries between different colors (see Figure 2.4 on page 75 for an example), not only the original colors are used to interpolate between, but also colors with reduced chroma, namely colors \check{c} and \check{c} with chroma reduced by a factor of $3/4$ and $2/3$, respectively.

Let $cd(p_0, c_+, c_-)$ be the minimum distance of c_0 from one of the lines $\overline{c_+c_-}$, $\overline{\check{c}_+\check{c}_-}$, and $\overline{\check{c}_+\check{c}_-}$. The color evidence is then computed as $e_c(p_0, c_+, c_-) := 1 - cd(p_0, c_+, c_-)/(0.4 \cdot \|c_+ - c_-\|)$. Let k be the number of directions for which c_+ is the color of the nearest classified pixel and c_- is the color of the nearest classified pixel in opposite direction, and let d be the minimum sum of distances in which c_+ and c_- appear in opposite directions. If $d \leq \theta_d + 1$, the position evidence $e_p(p_0, c_+, c_-)$ is computed as $1.0 - 0.25 \cdot (d - 2)/(\theta_d - 1) + 0.25 \cdot (k - 1)$ restricted to $[0, 1]$, which means that it is 1 if the distance is minimal or if there are at least two directions. The overall evidence $e_o(p_0, c_+, c_-)$ is computed as the product of color and of position evidence.

If $e_o(p_0, c_+, c_-) > 0.5$, the pixel is candidate for being assigned to one of the two clusters. Let $d_+ = \|c_0 - c_+\|$ and $d_- = \|c_0 - c_-\|$ be the color distances and n_+ and n_- be the number of neighbors having color c_+ and c_- respectively. The priority for assigning p_0 color c_+ is computed as $d_+/(d_+ + d_-) - 0.05 \cdot n_+$ and the priority for assigning it color c_- is computed accordingly. Iteratively, the pixel with the overall smallest priority value is assigned to the corresponding cluster and the priorities of the neighboring pixels are updated.

10th step: aggressive assignment The priority for a pixel to be added to a cluster is computed as the distance between the pixel's and the cluster's colors, which—depending on the number of common edges—

³ Here, the *chroma* of a color is its distance from the nearest gray tone in RGB color space. Reducing the chroma corresponds to simultaneously reducing saturation and value in HSV color space.

is decreased by a factor between 1 and 1/2, and—depending on the cluster’s ditheredness—is decreased by a factor between 1 and 1/4, and—depending on the pixel’s ditheredness—is decreased by a factor between 1 and 1/2, and—depending on the pixels’s closeness to a boundary between very different colors—is decreased by a factor between 1 and 1/2. The threshold on the priority is set to 25.

11th step: merging clusters based on color again The priority for two clusters to be merged is computed as the distance of the clusters’ colors, which—depending on the clusters’ ditheredness—is decreased by a factor between 1 and 1/2. The threshold on the priority is set to 25.

12th step: clustering rest The colors of the pixels are subdivided using a regular grid.⁴ According to the distance to the colors of existing clusters, every grid cell gets a weight. The mean color of the pixels in the cell with the largest weighted number of votes defines a new cluster, and all pixels with a color that has a sufficiently small difference to this new cluster’s color are assigned to it.

The grids used to partition color space are based on cubes of sidelength 16, one grid starting at the origin (black), one grid starting at (8, 8, 8). As long as there are still unclassified pixels, the grid cell with the highest product of number of pixels with color in it and minimum distance from center to any existing cluster’s color is chosen as basis for a new cluster: The average color of the pixels that have distance less than 8 from the cell’s center is the new clusters color, and all pixels having distance smaller than 50 to this color are classified.

Merging of Small Shapes

Textured Regions For the detection of regions of high complexity, the side length d_b of the rectangular function used as low pass filter is set to $2 \cdot s(I)/25 + 5.5$ and the height is set to $1/(d_b \cdot d_b)$ such that the sum of all brightness values in the image stays unchanged. In order to encounter, that for small images the edge density of regions containing shapes that are perceived as individual items may be significantly higher than for larger images, the threshold on the brightness is set to $\theta_t = 255 \cdot (0.14 + 7/s(I))$. On the one hand this ensures that for images with side lengths smaller than 8, the threshold is larger than the maximum possible value and thus even groups of shapes consisting of single pixels will not be classified as texture. On the other hand, the threshold is in the range 0.15 ± 0.01 for images with side lengths larger than 350 which yields

⁴ In order to reduce unwanted quantization effects, actually two grids are used such that the cells overlap.

B. Specification of Parameter Values

good results in practice. The radius of the disk used for the closing and the opening is set to $d_b/4 = 0.5 \cdot s(I)/25 + 5.5$. For the extraction of textured shapes, the side length of the low pass filter is set to $d_b/5$, the threshold on the brightness is left unchanged, and the radius of the disk used for the closing and the opening is set to $d_b/10$.

Broken Lines For the reconstruction of line shapes, the weighted distance between two points p_{i_1} and p_{j_1} belonging to the chains $P_i = (p_{i_1}, \dots, p_{i_k})$ and $P_j = (p_{j_1}, \dots, p_{j_l})$, respectively is determined based on the following features:

distance The Euclidean distance $\|p_{i_1} - p_{j_1}\|$ between p_{i_1} and p_{j_1} . The closer the points, the higher the likeliness of being consecutive points on some curve.

distribution of nearest neighbors For every point the distribution of points in its neighborhood is analyzed. For a point that actually does originate from the depiction of a line shape, there are probably also other points from that line shape contained in the neighborhood, situated in a rather thin corridor containing the point itself. For a point that originates from noise, on the other hand, the points in the neighborhood will usually be scattered or—if the noisy point lies near a line shape passing by—will be situated in a rather thin corridor *not* containing the point. The likeliness of originating from a line shape is estimated using the idea of *probabilistic relaxation*: The maximally consistent assignment of labels to objects is determined by propagating local clues (see, e. g., [196] and [140]).

Every point p is assigned a direction v , and a value b indicating the belief that v corresponds to the course of a line shape that the point originates from. Initially v and b are computed using principal component analysis (as introduced in [180]) of the nearest neighbors of p . In several rounds, the directions as well as the beliefs are updated based on consistencies of the directions of pairs of points. The higher the beliefs of two points, the higher the likeliness of being consecutive points on some curve.

direction The direction of the edge between points p_{i_1} and p_{j_1} in relation to the courses of the chains P_i and P_j . Due to the discrete nature of the original polygonal shape and its skeleton, consecutive edges of a correct reconstruction might pretty well form a right angle. The direction of an edge, therefore, is not rated on the direction of a single predecesing edge only. The longer a potential edge e , the longer the part of the already reconstructed curve supporting the assumption that e belongs to the curve should be.

For the edge $e = \overline{p_{i_1} p_{j_1}}$ of length d connecting chains P_i and P_j , let $\tilde{p}(e, P_i)$ be the point on the polyline P_i with geodesic distance d from p_{i_1} (if the length of P_i is smaller than d , then $\tilde{p}(e, P_i) := p_{i_k}$). The degree to which a part of P_i points into the direction of e may be estimated using the length of its projection to e . The support from P_i for the edge e is therefore determined as $s(e, P_i) := \langle p_{i_1} - \tilde{p}(e, P_i), (p_{j_1} - p_{i_1})/d \rangle$ (see Figure B.1 for an illustration). The support from P_j for the edge e is determined analogously. The greater the support for an edge, the higher the likeliness that it connects consecutive points on some curve.

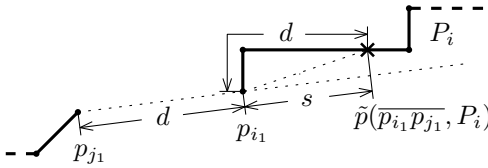


Figure B.1. Direction of edges and curves: notations used in the definition of the directional support.

goodness of form The goodness of form of the resulting chain. Human perception tends to prefer simple figures over complex ones (see Section 1.2). Edges leading to curves of low visual complexity therefore are preferred over edges leading to erratic structures. The visual complexity is estimated based on two features, namely the degree to which the part of the resulting chain near the new edge is close to a straight line⁵, and the ratio of the total length of the curve and its diameter. The closer to a straight line the resulting chain is and the smaller the ratio of length and diameter, the better the form. The better the form of the chain resulting from connecting two points, the higher the likeliness that these points are consecutive points on some curve.

coverage The density of points on the resulting chain or more formally, the ratio of the chain's parts covered by shape (in contrast to the gaps). The more of the line is actually depicted, the longer the gap between two parts might be. In dashed lines, e.g., the gaps may surely be larger than in dotted lines without destroying the perception of a line—especially in the presence of noise. An approximation of the covered parts can be determined based on the vertices of the chain and the side lengths of the corresponding squares as determined during the computation of

⁵ Please note that in the analysis of the *distribution of nearest neighbors* also the degree of closeness to a straight line was rated, but there all points in a spacial neighborhood of a point were considered, here only the points forming the chain are considered

B. Specification of Parameter Values

the max- L_∞ -skeleton. For a point p of the max- L_∞ -skeleton of a shape, let $r(p)$ be half the side length of the corresponding square. The covered part of a chain $P = (p_1, \dots, p_m)$ is estimated as $\sum_{i=0}^{m-1} \max(\|p_{i+1} - p_i\|, \sqrt{2}(r(p_i) + r(p_{i+1})))$. Dividing this value by the chain's length gives the relative coverage. The higher the coverage of the chain resulting from connecting two points, the higher the likeliness that these points are consecutive points on some curve.

Simplification

Polyline Simplification For the *adaptive coarsening plus corner simplification* the error threshold Δ_{break} to stop further merges is set to $4 \cdot \Delta_{max}$ and the factor ϱ_{ac} determining the error tolerance is set to 0.1. The minimum angle between edges to be considered for corner simplification is set to 10° , the factor ϱ^{cc} determining the maximum distance of the tip is set to 0.1, and the width ε^{cc} of the additional neighborhood around the tip triangle is set to $\min(\Delta_{max}, r)$.

B.2. Similarity Estimation based on Mapping

This section gives a set of suitable values for the parameters of the comparison of shape sets as described in Chapter 3.

Let $s(I_1)$ be the image size of image I_1 and $s(I_2)$ be the image size of image I_2 , respectively. For computing a similarity transformation, in case of a complete-complete comparison the thresholds for the minimum and the maximum scaling factor are set to $c^{min} = 0.25 \cdot s(I_2)/s(I_1)$ and $c^{max} = 4 \cdot s(I_2)/s(I_1)$. In case of a complete-partial or a complete-semi-partial comparison the thresholds for the minimum and the maximum scaling factor are set to $c^{min} = 0.02 \cdot s(I_2)/s(I_1)$ and $c^{max} = 2 \cdot s(I_2)/s(I_1)$. The additional random scaling factor c_r is chosen such that $\log_2(c_r)$ is normal distributed with standard deviation 0.1. The threshold θ^a for deciding whether two vertices are identified is set to 0.2. The threshold θ^e for the relative error still tolerated during the augmentation of a sample is set to 0.1.

In case of a complete-complete comparison the exponents used in the computation of a samples weight are set to $e_1 = 1$ and $e_2 = 1$. In case of a complete-partial or a complete-semi-partial comparison the exponents are set to $e_1 = 2$ and $e_2 = 1$. The factor $\omega_{V,min}$ of maximal penalization of samples not spanning two dimensions is set to 0.25. The factor c_E for adjusting the tolerance against errors is set to 5.

The cluster radius r is set to $0.05 \cdot s(I_2)$.

The threshold d_{max} on the distance in the computation of the *substitution similarity* is set to $0.2 \cdot s(I_2)$.

B.3. Similarity Estimation based on Image Primitives

This section gives a set of suitable values for the parameters of the comparison of shape sets as described in Section 4.4.

Extraction of Figures Given a single polyline P , whether and how it may be represented by one of the primitives is decided as described in the following:

rectangle Let P' be the polyline resulting from applying the *coarsening plus corner simplification* to (a subdivision of) P , using 0.1 times the diameter of P as error threshold. For every 4-subsequence of edges in P' such that the angles of the supporting lines differ less than 10° from the respective angles in a rectangle, the polyline P'' resulting from the intersections of the supporting lines is constructed. If the *substitution similarity* $\Phi^{cc}(\{P''\}, \{P\})$ is sufficiently large, the rectangle generated from P'' is assumed to be a good representation of P .

ellipse By principal component analysis of P , the direction \vec{v} of the major axis of a potential ellipse is determined. The maximum extents of P in direction of \vec{v} and perpendicular to \vec{v} determine the first approximating ellipse E . Using 100 equally spaced sample points on P , the best similarity transformation t mapping corresponding points from E to the points on P is computed according to Section 3.2.4. Let E' be the ellipse resulting from applying t to E . If the *substitution similarity* $\Phi^{cc}(\{E'\}, \{P\})$ is sufficiently large, E' is assumed to be a good elliptical representation of P .

triangle Let P' be the polyline resulting from applying the *coarsening plus corner simplification* to (a subdivision of) P , using 0.1 times the diameter of P as error threshold. For every 3-subsequence of edges in P' such that the supporting lines form a triangle with smallest angle greater than 10° , the polyline P'' resulting from the intersections of the supporting lines is constructed. If the *substitution similarity* $\Phi^{cc}(\{P''\}, \{P\})$ is sufficiently large, P'' is assumed to be a good triangular representation of P .

B. Specification of Parameter Values

Given a set \mathcal{P} of polylines, whether one of the basic primitives can be extracted is decided based on comparing a square, a circle and an equilateral triangle, respectively, to \mathcal{P} using mapping under similarity transformations and determining the directed *substitution similarity* Φ^r according to Chapter 3.

In the clustering of polylines the threshold on the distance is primary set to $\theta_c := 0.2 \cdot s(I)$, but whenever the number of resulting clusters exceeds the envisaged number of figures it is adjusted to $\theta_c := \theta_c \cdot 1.2$ and the polylines are reclustered. Small clusters (diameter smaller than 0.1 times the diameter of the largest figure extracted so far) are deleted.

For deciding which way of decomposing a given cluster \mathcal{C} should be chosen, for each of the four options a support value is computed. The support for a figure F_{ch} derived from the convex hull is determined as *substitution similarity* $\Phi^r(F_{ch}, \mathcal{C})$ times a factor favoring figures with aspect ratio close to 1. If there is no polyline in \mathcal{C} that nearly equals the convex hull, the support for a basic geometric figure F_a at arbitrary position is determined in the same way. The support for the representation by the individual polylines is determined as 0.9 times the average over the individual factors for the aspect ratios. For F_{ch} and F_a , the part \mathcal{C}' of \mathcal{C} that is not represented by the figure, is computed based on an analysis of the resemblance function ϕ in the computation of the *substitution similarity* $\Phi^r(\mathcal{C}, F)$.

If the number of resulting figures (For F_{ch} or F_a , respectively, it is 1 in case that $\mathcal{C}' = \emptyset$ and 2 otherwise. For representation by individual polylines it is $|\mathcal{C}'|$.) does not exceed the envisaged number of figures (which is 8 minus the number of figures already extracted, minus the number of other clusters), and the support is sufficiently large, the best of these representations is chosen. Otherwise the cluster is represented by \mathcal{C} as a single figure.

The conflation of ‘concentric’ figures is done based on scaling with respect to the center of mass. If the two figures are of the same type, if the *substitution similarity* of the scaled smaller figure and the larger figure is sufficiently large, and if the scaling factor is not smaller than 0.5, then the two figures are conflated.

A figure’s degree of being a frame is determined depending on the figure’s shape and on its position relative to the other figures. A convex figure with all other figures lying in the inside and an aspect ratio between 1/1.5 and 1.5 gets a value of 1. A convex figure with all other figures except the first frame lying in the inside and an aspect ratio between 1/1.5 and 1.5 gets a value of 0.5. In both cases, if the aspect ratio’s deviation from 1 is greater, the value is decreased.

Given a pair (F_i, F_j) of figures, the relation $R_{i,j}$ stores the relative size $s(i, j)$, the relative distance $d(i, j)$ which is basically the distance of the centers divided

B.3. Similarity Estimation based on Image Primitives

by the sum of the sizes, and information about their similarity: $sim_t(i, j)$ is the average of the *substitution similarity* values when F_i and F_j are translated such that their centers coincide and when they are scaled and translated such that their centers and their sizes coincide; $sim_r(i, j)$ is the average of the respective two values under rigid motions and $sim_m(i, j)$ is the average of the respective two values under reflections.

Comparison of Images The importance $\omega'(F_i)$ of a figure F_i mainly depends on the size, but—in order not to disregard small figures too much—sublinearly. It is set to the 1.5th root of the relative size times a factor between 0.5 and 1 depending on the figure’s degree of being a frame, times a factor for its uniqueness: If there are several figures having the same shape (as indicated by the value $sim_t(i, j)$), for every such figure the importance of subsequent copies is set to 0.5 times their prior value. This means that for 8 identical figures, the importance of the 8th one is only 1/128 times its original value. The importance $\omega'(R_{i,j})$ of a relation $R_{i,j}$ between figures F_i and F_j is set to the average importance of F_i and F_j times a factor between 0.5 and 1 depending on the relative distance of the figures.

For the comparison of images, the weight ratio for figures and for relations is set to $\omega_{\mathcal{F}} = 0.9$ and $\omega_{\mathcal{R}} = 0.1$, respectively. The weights are then computed as $\omega(F_i) := \omega_{\mathcal{F}} \cdot \omega'(F_i) / \sum_{F \in \mathcal{F}} \omega'(F)$ and $\omega(R_{i,j}) := \omega_{\mathcal{R}} \cdot \omega'(R_{i,j}) / \sum_{R \in \mathcal{R}} \omega'(R)$.

The similarity value $\sigma_{\mathcal{F}}(F_{1_i}, F_{2_{i'}})$ is basically determined as $\Phi^{cc}(F_{1_i}, F_{2_{i'}})$. However, for the basic primitives *rectangle*, *ellipse*, and *triangle* predefined values are used. The similarity value $\sigma_{\mathcal{R}}(R_{1_i,1_j}, R_{2_{i'},2_{j'}})$ is composed of a factor between 0.5 and 1 depending on the analogy of the values sim_t , sim_r , and sim_m , a factor between 0.5 and 1 depending on the analogy of relative sizes, and a factor between 0.5 and 1 depending on the analogy of relative distances.

Bibliography

- [1] P. K. Agarwal and M. Sharir. Davenport-Schinzel sequences and their geometric applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 1–47. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000. [146](#)
- [2] P. K. Agarwal, M. Sharir, and S. Toledo. Applications of parametric searching in geometric optimization. In *SODA '92: Proceedings of the 3rd annual ACM-SIAM symposium on Discrete algorithms*, pages 72–82, Philadelphia, PA, USA, 1992. Society for Industrial and Applied Mathematics. [49](#)
- [3] P. K. Agarwal, S. Har-Peled, N. H. Mustafa, and Y. Wang. Near-linear time approximation algorithms for curve simplification. In *ESA '02: Proceedings of the 10th Annual European Symposium on Algorithms*, pages 29–41, London, UK, 2002. Springer. [92](#)
- [4] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner. A novel type of skeleton for polygons. *Journal of Universal Computer Science*, 1(12):752–761, 1995. [84](#), [142](#)
- [5] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 5:75–91, 1995. [50](#)
- [6] H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, 13:251–266, 1995. [48](#)
- [7] H. Alt, C. Knauer, and C. Wenk. Matching polygonal curves with respect to the Fréchet distance. In *Proceedings of the 18th International Symposium on Theoretical Aspects of Computer Science*, pages 63–74, 2001. [50](#)
- [8] H. Alt, L. Scharf, and D. Schymura. Probabilistic matching of planar regions. *Computational Geometry, Theory and Applications (CGTA)*, 43(2):99–114, 2010. Special Issue on the 24th European Workshop on Computational Geometry (EuroCG'08). [119](#)

Bibliography

- [9] T. D. Alter and D. W. Jacobs. Uncertainty propagation in model-based recognition. *International Journal of Computer Vision*, 27(2):127–159, 1998. [119](#)
- [10] S. Alwis and J. Austin. A novel architecture for trademark image retrieval systems. In *Challenge of Image Retrieval: Proceedings of the Electronic Workshops in Computing*, Newcastle upon Tyne, 1998. [139](#)
- [11] S. Alwis and J. Austin. A neural network architecture for trademark image retrieval. In J. Mira and J. Sánchez-Andrés, editors, *Engineering Applications of Bio-Inspired Artificial Neural Networks*, volume 1607 of *Lecture Notes in Computer Science*, pages 361–372. Springer Berlin/Heidelberg, 1999. [155](#)
- [12] N. Amenta, M. Bern, and D. Eppstein. The crust and the β -skeleton: Combinatorial curve reconstruction. In *Graphical Models and Image Processing*, pages 125–135, 1998. [81](#)
- [13] M. Arevalillo-Herráez, J. Domingo, and M. Zacarés. Probabilistic normalization: an approach to normalizing similarity measures in content based image retrieval. In *Proceeding of the 5th IASTED International Conference on Signal Processing, Pattern Recognition and Applications*, pages 30–35, Innsbruck, Austria, 2008. [54](#)
- [14] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. In *SODA '90: Proceedings of the 1st annual ACM-SIAM symposium on Discrete algorithms*, pages 129–137, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics. [141](#)
- [15] F. G. Ashby and N. A. Perrin. Toward a unified theory of similarity and recognition. *Psychological Review*, 95(1):124–150, 1988. [36](#), [37](#), [39](#), [41](#), [42](#), [46](#), [51](#)
- [16] F. Attneave. Dimensions of similarity. *American Journal of Psychology*, 63(4):516–556, 1950. [v](#), [37](#), [38](#), [43](#), [45](#), [51](#)
- [17] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954. [25](#), [26](#), [52](#), [121](#)
- [18] X. Bai, L. J. Latecki, and W. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):449–462, 2007. [84](#), [142](#)
- [19] X. Bai, X. Yang, L. J. Latecki, W. Liu, and Z. Tu. Learning context-sensitive shape similarity by graph transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):861–874, 2010. [149](#)

- [20] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981. 118
- [21] L. W. Barsalou. Construal in perception. URL http://userwww.service.emory.edu/~barsalou/Courses/Cognition/Lecture_Notes/T8a-4g-construal_perception-OUT.pdf. 29
- [22] L. W. Barsalou. Context-independent and context-dependent information in concepts. *Memory and Cognition*, 10(1):82–93, 1982. 39
- [23] S. O. Belkasima, M. Shridhara, and M. Ahmadi. Pattern recognition with moment invariants: a comparative study and new results. *Pattern Recognition*, 24(12):1117–1138, 1991. 140
- [24] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002. 157
- [25] M. de Berg, M. van Kreveld, and S. Schirra. A new approach to subdivision simplification. In *Auto-Carto 12 Proceedings of the International Symposium on Computer-Assisted Cartography*, pages 79–88. Cartography and Geographic Information Society, 1995. 91
- [26] F. Bernardini and C. L. Bajaj. Sampling and reconstructing manifolds using alpha-shapes. In *Proceedings of the 9th Canadian Conference on Computational Geometry*, pages 193–198, 1997. 81
- [27] S. Beucher and C. Lantuejoul. Use of watersheds in contour detection. In *International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation*, 1979. 70
- [28] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987. 165
- [29] I. Biederman and G. Ju. Surface versus edge-based determinants of visual recognition. *Cognitive Psychology*, 20:38–64, 1988. 26
- [30] I. Biederman, H. J. Hilton, and J. E. Hummel. Pattern goodness and pattern recognition. In J. R. Pomerantz and G. R. Lockhead, editors, *The Perception of Structure*, chapter 5, pages 73–95. APA, Washington, D.C., 1991. 25
- [31] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967. 81, 84, 142

Bibliography

- [32] M. Bober, J. D. Kim, H. K. Kim, Y. S. Kim, W.-Y. Kim, and K. Mueller. Summary of the results in shape descriptor core experiment. ISO/IEC JTC1/SC29/WG11/MPEG99/M4869, 1999. [63](#)
- [33] F. L. Bookstein. Size and shape spaces for landmark data in two dimensions. *Statistical Science*, 1(2):181–222, 1986. [117](#)
- [34] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965. [19](#)
- [35] V. Bruce, P. R. Green, and M. A. Georgeson. *Visual perception: Physiology, psychology and ecology*. Psychology Press, New York, fourth edition, 2003. [27](#)
- [36] M. Buchin. *On the Computability of the Frechet Distance Between Triangulated Surfaces*. PhD thesis, Freie Universität, Berlin, 2007. [49](#)
- [37] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, Athens, Greece, 2000. [55](#)
- [38] G. M. Campbell and R. G. Cromley. Optimal simplification of cartographic lines using shortest-path formulations. *The Journal of the Operational Research Society*, 42(9):793–802, 1991. [91](#)
- [39] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. [70](#)
- [40] T. A. Cass. Feature matching for object localization in the presence of uncertainty. Technical Report A.I. Memo No. 1133, Massachusetts Institute of Technology, 1990. [131](#)
- [41] A. Cerri, M. Ferri, and D. Giorgi. Retrieval of trademark images by means of size functions. *Graphical Models*, 68(5–6):451–471, 2006. [157](#)
- [42] A. Chalechale, A. Mertins, and G. Naghdy. Edge image description using angular radial partitioning. *IEE Proceedings of Vision, Image and Signal Processing*, 151(2):93–101, 2004. [140](#)
- [43] J. H. Challis. Estimation of the finite center of rotation in planar movements. *Medical Engineering & Physics*, 23(3):227–233, 2001. [126](#)
- [44] W. S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *Computational Geometry & Applications*, 6(1):59–77, 1996. [93](#)

- [45] D. Chang and K. V. Nesbitt. Developing gestalt-based design guidelines for multi-sensory displays. In *MMUI '05: Proceedings of the 2005 NICTA-HCSNet Multimodal User Interaction Workshop*, pages 9–16, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc. [29](#)
- [46] E. Y. Chang. Learning and measuring perceptual similarity. In *MMCBIR '01: Proceedings of the international workshop on MultiMedia Content Based Indexing and Retrieval*, pages 71–74, 2001. [62](#)
- [47] M.-T. Chang and S.-Y. Chen. Deformed trademark retrieval based on 2d pseudo-hidden markov model. *Pattern Recognition*, 34(5):953–967, 2001. [156](#)
- [48] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001. [34](#)
- [49] C.-K. Chen, Q.-Q. Sun, and J.-Y. Yang. Binary trademark image retrieval using region orientation information entropy. In *Proceedings of the International Conference on Computational Intelligence and Security Workshops*, pages 295–298, Washington, DC, USA, 2007. IEEE Computer Society. [158](#)
- [50] O. Cheong, J. Gudmundsson, H.-S. Kim, D. Schymura, and F. Stehn. Measuring the similarity of geometric graphs. In *Proceedings of the 8th International Symposium on Experimental Algorithms*, volume 5526 of *Lecture Notes in Computer Science*, pages 101–112, Dortmund, Germany, 2009. Springer. [148](#)
- [51] L. P. Chew and K. Kedem. Improvements on geometric pattern matching problems. In *Proceedings of the 3rd Scandinavian Workshop on Algorithm Theory*, pages 318–325. Springer, 1992. [49](#)
- [52] G. Ciocca and R. Schettini. Similarity retrieval of trademark images. In *Proceedings of the 10th International Conference on Image Analysis and Processing*, pages 915–920, 1999. [140](#), [156](#)
- [53] G. Ciocca and R. Schettini. Content-based similarity retrieval of trademarks using relevance feedback. *Pattern Recognition*, 34(8):1639–1655, 2001. [156](#)
- [54] K. L. Clarkson. Nearest-neighbor searching and metric space dimensions. In G. Shakhnarovich, T. Darrell, and P. Indyk, editors, *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*, pages 15–59. MIT Press, 2006. [33](#), [136](#)

Bibliography

- [55] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *ICCV '99: Proceedings of the International Conference on Computer Vision - Volume 2*, pages 1197–1203, Washington, DC, USA, 1999. IEEE Computer Society. [70](#)
- [56] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, third edition, 2009. [60](#)
- [57] G. Cortelazzo, G. A. Mian, G. Vezzi, and P. Zamperoni. Trademark shapes description by string-matching techniques. *Pattern Recognition*, 27(8):1005–1018, 1994. [154](#)
- [58] W. H. E. Day and H. Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1:1–24, 1984. [132](#)
- [59] B. N. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, 7(6):793–800, 1934. [88](#)
- [60] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2):101–124, 1993. [67](#)
- [61] A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, volume 34 of *Interdisciplinary Applied Mathematics*. Springer, New York, 2008. [26](#)
- [62] T. K. Dey and P. Kumar. A simple provable algorithm for curve reconstruction. In *SODA '99: Proceedings of the 10th annual ACM-SIAM symposium on Discrete algorithms*, pages 893–894, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics. [82](#)
- [63] T. K. Dey, K. Mehlhorn, and E. A. Ramos. Curve reconstruction: connecting dots with good reason. *Computational Geometry*, 15(4):229–244, 2000. [81](#), [83](#)
- [64] DIN 6174:2007-10. Colorimetric evaluation of colour coordinates and colour differences according to the approximately uniform CIELAB colour space, 2007. [187](#)
- [65] D. Dori, L. Wenying, and M. Peleg. How to win a dashed line detection contest. In R. Kasturi and K. Tomre, editors, *Graphics Recognition Methods and Applications*, volume 1072 of *Lecture Notes in Computer Science*, pages 286–300. Springer Berlin/Heidelberg, 1996. [81](#)
- [66] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973. [94](#)

- [67] M.-P. Dubuisson and A. K. Jain. A modified Hausdorff distance for object matching. In *Proceedings of the 12th International Conference on Pattern Recognition*, volume 1, pages 566–568. IEEE Computer Society Press, 1994. 138
- [68] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972. 81, 118
- [69] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973. 161
- [70] M. C. Dyson and H. Box. Retrieving symbols from a database by their graphic characteristics: Are users consistent? *Journal of Visual Languages & Computing*, 8(1):85–107, 1997. 28
- [71] J. P. Eakins, K. Shields, and J. Boardman. ARTISAN – a shape retrieval system based on boundary family indexing. In *Storage and Retrieval for Still Image and Video Databases IV. Proceedings SPIE 2670*, pages 17–28, 1996. 101, 159
- [72] J. P. Eakins, J. M. Boardman, and M. E. Graham. Similarity retrieval of trademark images. *IEEE MultiMedia*, 5(2):53–63, 1998. 55, 64, 65, 150, 159, 163, 169
- [73] J. P. Eakins, J. D. Edwards, K. J. Riley, and P. L. Rosin. Comparison of the effectiveness of alternative feature sets in shape retrieval of multicomponent images. In *Proceedings of the Conference on Storage and Retrieval for Media Databases*, volume 4315, pages 196–207. SPIE, 2001. 77, 154, 159
- [74] J. P. Eakins, K. J. Riley, and J. D. Edwards. Shape feature matching for trademark image retrieval. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 28–38, 2003. 150, 159, 163, 164, 169
- [75] S. Edelman, N. Intrator, and T. Poggio. Complex cells and object recognition. Unpublished manuscript, 1997. URL <http://kybele.psych.cornell.edu/~edelman/Archive/nips97.pdf>. 13
- [76] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983. 81
- [77] R. Estkowski and J. S. B. Mitchell. Simplifying a polygonal subdivision while keeping it simple. In *SCG '01: Proceedings of the 17th annual symposium on Computational geometry*, pages 40–49, New York, NY, USA, 2001. ACM. 91

Bibliography

- [78] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3):231–262, 1994. [154](#)
- [79] G. T. Fechner. *Elemente der Psychophysik*. Breitkopf und Hartel, 1860. [24](#), [45](#)
- [80] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. [70](#)
- [81] P. F. Felzenszwalb and J. D. Schwartz. Hierarchical matching of deformable shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. [149](#)
- [82] L. H. de Figueiredo and J. de Miranda Gomes. Computational morphology of curves. *The Visual Computer*, 11:105–112, 1994. [81](#)
- [83] S. Fillenbaum and A. Rapoport. Verbs of judging, judged: A case study. *Journal of Verbal Learning and Verbal Behavior*, 13(1):54–62, 1974. [38](#)
- [84] 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(6):381–395, 1981. [81](#), [117](#), [119](#)
- [85] M. M. Fisher and K. Smith-Gratto. Gestalt theory: A foundation for instructional screen design. *Journal of Educational Technology Systems*, 27(4), 1998/1999. [29](#)
- [86] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *Computer*, 28: 23–32, 1995. [154](#)
- [87] R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial gray scale. In *Proceedings of the Society for Information Display*, volume 17, pages 75–77, 1976. [19](#)
- [88] J. Flusser, J. Kautsky, and F. Šroubek. Object recognition by implicit invariants. In *CAIP'07: Proceedings of the 12th international conference on Computer analysis of images and patterns*, pages 856–863, Berlin/Heidelberg, 2007. Springer. [140](#)
- [89] M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo*, 22(1):1–72, 1906. [49](#)

- [90] K. S. Fu. *Syntactic pattern recognition and applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982. 165
- [91] J. A. García and J. Fdez-Valdivia. Boundary simplification in cartography preserving the characteristics of the shape features. *Computers & Geosciences*, 20(3):349–368, 1994. 95
- [92] I. Gati and A. Tversky. Weighting common and distinctive features in perceptual and conceptual judgments. *Cognitive Psychology*, 16(3):341–370, 1984. 37
- [93] C. F. Gauß. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium (Theorie der Bewegung der Himmelskörper, welche in Kegelschnitten die Sonne umlaufen)*. F. Perthes and I. H. Besser (C. Meyer), Hamburg, 1809 (Hannover, 1865). 124
- [94] J. J. Gibson. *The perception of the visual world*. Houghton Mifflin, Boston, 1950. 26
- [95] M. Godau. *On the Complexity of Measuring the Similarity between Geometric Objects in Higher Dimensions*. PhD thesis, Freie Universität, Berlin, 1998. 49
- [96] E. Goldmeier. Über Ähnlichkeit bei gesehenen Figuren. *Psychologische Forschung*, 21:146–208, 1937. 24, 27, 38, 44, 45, 52
- [97] R. L. Goldstone. Mainstream and avant-garde similarity. *Psychologica Belgica*, 35:145–165, 1995. 40
- [98] R. L. Goldstone. Learning to perceive while perceiving to learn. In R. Kimchi, M. Behrmann, and C. R. Olson, editors, *Perceptual organization in vision: Behavioral and neural perspectives*, pages 233–278. Lawrence Erlbaum Associates, New Jersey, 2003. 26
- [99] J. E. Goodman and J. O'Rourke, editors. *Handbook of discrete and computational geometry*. CRC Press, Inc., Boca Raton, FL, USA, 1997. 19
- [100] Google Images. URL <http://images.google.com>. 56
- [101] R. L. Gregory. *The Intelligent Eye*. Weidenfeld and Nicolson, London, 1970. 30
- [102] A. Gribov and E. Bodansky. A new method of polyline approximation. In A. Fred, T. Caelli, R. P. W. Duin, A. Campilho, and D. de Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition - Proceedings of the Joint IAPR International Workshops, SSPR 2004 and SPR 2004*, pages 504–511. Springer, 2004. 96

Bibliography

- [103] M. Hagedoorn and R. C. Veltkamp. Measuring resemblance of complex patterns. In G. Bertrand, M. Couprie, and L. Perroton, editors, *Discrete Geometry for Computer Imagery*, volume 1568 of *Lecture Notes in Computer Science*, pages 286–297. Springer Berlin/Heidelberg, 1999. [138](#)
- [104] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132, 1985. [70](#)
- [105] C. G. Healey, R. S. Amant, and M. S. Elhaddad. ViA: a perceptual visualization assistant. In W. R. Oliver, editor, *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*, volume 3905, pages 2–11. SPIE, 2000. [30](#)
- [106] H. J. A. M. Heijmans and C. Ronse. The algebraic basis of mathematical morphology. I. dilations and erosions. *Computer Vision, Graphics, and Image Processing*, 50(3):245–295, 1990. [79](#)
- [107] H. von Helmholtz. *Handbuch der physiologischen Optik*, volume 9 of *Encyklopädie der Physik*. Leopold Voss, Leipzig, 1867. [26](#)
- [108] J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Information Processing Letters*, 33(4):169–174, 1989. [146](#)
- [109] J. Hershberger and J. Snoeyink. Speeding up the Douglas-Peucker line-simplification algorithm. In *Proceedings of the 5th International Symposium on Spatial Data Handling*, volume 1, pages 134–143, Charleston, South Carolina, 1992. [94](#), [98](#)
- [110] C. A. R. Hoare. Quicksort. *The Computer Journal*, 5(1):10–16, 1962. [60](#)
- [111] J. E. Hochberg. *Perception*. Foundations of Modern Psychology. Prentice-Hall, Englewood Cliffs, New Jersey, second edition, 1978. [25](#), [26](#)
- [112] V. J. Hodge, G. Hollier, J. P. Eakins, and J. Austin. Eliciting perceptual ground truth for image segmentation. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 320–329, 2006. [28](#)
- [113] V. J. Hodge, G. Hollier, J. Austin, and J. P. Eakins. Identifying perceptual structures in trademark images. In *Proceedings of the 5th IASTED International Conference on Signal Processing, Pattern Recognition and Applications*, pages 81–86, Anaheim, CA, USA, 2008. ACTA Press. [77](#), [110](#), [188](#)

- [114] Z. Hong and Q. Jiang. Hybrid content-based trademark retrieval using region and contour features. In *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops*, pages 1163–1168, Washington, DC, USA, 2008. IEEE Computer Society. [115](#), [158](#)
- [115] P. V. C. Hough. Methods and means for recognizing complex patterns, 1962. United States Patent 3,069,654, 1962-12-18. [81](#), [118](#)
- [116] D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proceedings of the 1st International Conference on Computer Vision*, pages 102–111, 1987. [116](#)
- [117] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990. [116](#)
- [118] D. P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of Voronoi surfaces and its applications. *Discrete and Computational Geometry*, 9(1):267–291, 1993. [49](#)
- [119] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993. [49](#), [138](#)
- [120] H. Imai and M. Iri. Polygonal approximations of a curve – formulations and algorithms. In G. T. Toussaint, editor, *Computational Morphology*, volume 6 of *Machine intelligence and pattern recognition*, pages 71–86. Elsevier Science Publishers, North Holland, 1988. [91](#), [92](#), [93](#)
- [121] S. Imai. Pattern similarity and cognitive transformations. *Acta Psychologica*, 41(6):433–447, 1977. [52](#)
- [122] S. Irani and P. Raghavan. Combinatorial and experimental results for randomized point matching algorithms. In *SCG '96: Proceedings of the 12th annual symposium on Computational geometry*, pages 68–77, New York, NY, USA, 1996. ACM. [117](#), [118](#)
- [123] ISO 128-20:1996. Technical drawings – general principles of presentation – part 20: Basic conventions for lines. International Organization for Standardization, 1996. [81](#)
- [124] ISO/IEC 15948:2004. Portable Network Graphics (PNG): Functional specification. International Organization for Standardization, 2004. [18](#)

Bibliography

- [125] ITU-T.81. Information technology – digital compression and coding of continuous-tone still images – requirements and guidelines. International Telecommunication Union (Telecommunication Standardization Sector), 1992. [18](#), [69](#)
- [126] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991. [77](#)
- [127] A. K. Jain and A. Vailaya. Shape-based retrieval: a case study with trademark image databases. *Pattern Recognition*, 31(9):1369–1390, 1998. [155](#), [156](#)
- [128] W. James. *The principles of psychology*, volume 1. Harvard, 1890. URL <http://psychclassics.yorku.ca/James/Principles/index.htm>. [38](#)
- [129] M. Jian and L. Xu. Trademark image retrieval using wavelet-based shape features. In *Proceedings of the International Symposium on Intelligent Information Technology Application Workshops*, pages 496–500, Los Alamitos, CA, USA, 2008. IEEE Computer Society. [158](#)
- [130] H. Jiang, C.-W. Ngo, and H.-K. Tan. Gestalt-based feature similarity measure in trademark database. *Pattern Recognition*, 39(5):988–1001, 2006. [140](#), [157](#), [159](#)
- [131] T. Kaneko. Line structure extraction from line-drawing images. *Pattern Recognition*, 25(9):963–973, 1992. [81](#)
- [132] F. Karamzadeh and M. A. Azgomi. An automated system for search and retrieval of trademarks. In *Proceedings of the 11th International Conference on Electronic Commerce*, pages 374–377, New York, NY, USA, 2009. ACM. [115](#), [158](#)
- [133] Y.-S. Kim and W.-Y. Kim. Content-based trademark retrieval system using visually salient feature. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 307–312, Los Alamitos, CA, USA, 1997. IEEE Computer Society. [155](#)
- [134] Y.-S. Kim, Y.-S. Kim, W.-Y. Kim, and M.-J. Kim. Development of content-based trademark retrieval system on the world wide web. *ETRI Journal*, 21(1):39–53, 1999. [155](#)
- [135] D. G. Kirkpatrick and J. D. Radke. A framework for computational morphology. In G. T. Toussaint, editor, *Computational Geometry, Machine Intelligence and Pattern Recognition*, pages 217–248. North-Holland, Amsterdam, 1985. [81](#)

- [136] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983. [61](#)
- [137] B. Kong, I. Phillips, R. Haralick, A. Prasad, and R. Kasturi. A benchmark: Performance evaluation of dashed-line detection algorithms. In R. Kasturi and K. Tombre, editors, *Graphics Recognition Methods and Applications*, volume 1072 of *Lecture Notes in Computer Science*, pages 270–285. Springer Berlin/Heidelberg, 1996. [81](#)
- [138] C. L. Krumhansl. Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. *Psychological Review*, 85(5):445–463, 1978. [37](#), [38](#), [51](#), [52](#)
- [139] J. B. Kruskal, Jr. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956. [86](#)
- [140] T. Kubota, T. Huntsberger, and J. T. Martin. Edge based probabilistic relaxation for sub-pixel contour extraction. In *EMMCVPR '01: Proceedings of the 3rd International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 328–343, London, UK, 2001. Springer. [194](#)
- [141] Y. Lamdan and H. J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision*, pages 238–249, 1988. [117](#)
- [142] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Object recognition by affine invariant matching. Technical Report 342, New York University, Courant Institute of Mathematical Sciences, 1988. [117](#)
- [143] L. J. Latecki, R. Lakämper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 424–429, 2000. [63](#), [149](#)
- [144] E. L. Lawler. The quadratic assignment problem. *Management Science*, 9(4):586–599, 1963. [168](#)
- [145] A.-M. Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*, chapter Sur la Méthode des moindres carrés. Firmin Didot, Paris, 1805. [124](#)
- [146] T. Lenz. How to sample and reconstruct curves with unusual features. In *Proceedings of the 22nd European Workshop on Computational Geometry (EWCG)*, Delphi, Greece, 2006. [82](#), [83](#)

Bibliography

- [147] T. Lenz. *Simple Reconstruction of Non-Simple Curves and Approximating the Median in Streams with Constant Storage*. PhD thesis, Freie Universität Berlin, 2008. [82](#)
- [148] R. H. van Leuken, M. F. Demirci, V. J. Hodge, J. Austin, and R. C. Veltkamp. Layout indexing of trademark images. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, pages 525–532, New York, NY, USA, 2007. ACM. [164](#)
- [149] W. H. Leung and T. Chen. Trademark retrieval using contour-skeleton stroke classification. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 517–520. IEEE, 2002. [156](#)
- [150] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. [52](#)
- [151] B. Li, E. Y. Chang, and Y. Wu. Discovery of a perceptual distance function for measuring image similarity. *Multimedia Systems*, 8(6):512–522, 2003. [47](#), [62](#)
- [152] L. Li, D. Wang, and G. Cui. Trademark image retrieval using region Zernike moments. In *Proceedings of the Workshop on Intelligent Information Technology Applications*, volume 2, pages 301–305, Los Alamitos, CA, USA, 2008. IEEE Computer Society. [158](#)
- [153] S. D. Lin, S.-C. Shie, W.-S. Chen, B. Y. Shu, X. L. Yang, and Y.-L. Su. Trademark image retrieval by distance-angle pair-wise histogram. *International Journal of Imaging Systems and Technology*, 15(2):103–113, 2005. [157](#)
- [154] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982. [132](#)
- [155] D. G. Lowe. Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image, 2004. United States Patent 6,711,293, 2004-03-23. [13](#), [67](#)
- [156] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. [67](#)
- [157] R. D. Luce. A psychophysical theory of intensity proportions, joint presentations, and matches. *Psychological Review*, 109:520–532, 2002. [24](#)
- [158] E. Mach. *Die Analyse der Empfindungen und das Verhältniss des Physischen zum Psychischen*. G. Fischer, Jena, fourth edition, 1903. [24](#), [45](#)

- [159] F. Mahmoudi, J. Shanbehzadeh, A.-M. Eftekhari-Moghadam, and H. Soltanian-Zadeh. Image retrieval based on shape similarity by edge orientation autocorrelogram. *Pattern Recognition*, 36(8):1725–1736, 2003. [140](#)
- [160] R. B. McMaster. A statistical analysis of mathematical measures for linear simplification. *The American Cartographer*, 13(2):103–116, 1986. [90](#)
- [161] R. B. McMaster and K. S. Shea. *Generalization in digital cartography*. Association of American Geographers, Washington, D.C., 1992. [90](#), [93](#)
- [162] D. L. Medin, R. L. Goldstone, and D. Gentner. Similarity involving attributes and relations: Judgments of similarity and difference are not inverses. *Psychological Science*, 1(1):64–69, 1990. [41](#)
- [163] P. M. Merlin and D. J. Farber. A parallel mechanism for detecting curves in pictures. *IEEE Transactions on Computers*, 24:96–98, 1975. [118](#)
- [164] G. A. Miller and J. A. Selfridge. Verbal context and the recall of meaningful material. *The American Journal of Psychology*, 63(2):176–185, 1950. [26](#)
- [165] 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. [119](#)
- [166] F. Mokhtarian and A. K. Mackworth. Scale-based description and recognition of planar curves and two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):34–43, 1986. [141](#)
- [167] MPEG. URL <http://www.chiariglione.org/mpeg>. [63](#)
- [168] D. B. Mumford. Mathematical theories of shape: Do they model perception? In *Proceedings of the Conference on Geometric Methods in Computer Vision*, volume 1570, pages 2–10. SPIE International Society for Optical Engineering, 1991. [25](#), [42](#), [53](#), [114](#)
- [169] D. B. Mumford. Pattern theory: the mathematics of perception. In *Proceedings of the International Congress of Mathematicians*, volume 1, pages 401–422. Higher Education Press (Beijing), 2002. [29](#)
- [170] D. B. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989. [70](#)

Bibliography

- [171] C. W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. QBIC project: querying images by content, using color, texture, and shape. In *Proceedings of the Conference on Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187. SPIE, 1993. [154](#)
- [172] F. Nielsen. *Visual Computing: Geometry, Graphics, And Vision (Graphics Series)*. Charles River Media, Inc., Rockland, MA, USA, June 2005. [117](#)
- [173] C. F. Olson. Time and space efficient pose clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 251–258, 1994. [119](#), [131](#)
- [174] H. P. Op de Beeck, K. Torfs, and J. Wagemans. Perceived shape similarity among unfamiliar objects and the organization of the human object vision pathway. *Journal of Neuroscience*, 28(40):10111–10123, 2008. [19](#)
- [175] J. O'Rourke and R. Washington. Curve similarity via signatures. In G. T. Toussaint, editor, *Computational Geometry*, Machine Intelligence and Pattern Recognition, pages 295–317. North-Holland, Amsterdam, 1985. [141](#)
- [176] J.-S. Park. Hierarchical shape description using skeletons. In T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua, and L.-T. Chia, editors, *Advances in Multimedia Modeling*, volume 4351 of *Lecture Notes in Computer Science*, pages 709–718. Springer Berlin/Heidelberg, 2007. [142](#)
- [177] T. Pavlidis. *Structural Pattern Recognition*. Springer, Berlin, Heidelberg, New York, 1977. [165](#)
- [178] T. Pavlidis. *Algorithms for graphics and image processing*. Springer, Berlin, 1982. [95](#)
- [179] T. Pavlidis and S. L. Horowitz. Segmentation of plane curves. *IEEE Transactions on Computers*, 23(8):860–870, 1974. [94](#)
- [180] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901. [130](#), [194](#)
- [181] H.-L. Peng and S.-Y. Chen. Trademark shape recognition using closed contours. *Pattern Recognition Letters*, 18(8):791–803, 1997. [155](#)
- [182] E. G. M. Petrakis, K. Kontis, E. Voutsakis, and E. E. Milios. Relevance feedback methods for logo and trademark image retrieval on the web. In *Proceedings of the Symposium on Applied Computing*, pages 1084–1088, New York, NY, USA, 2006. ACM. [157](#)

- [183] A. M. Pinheiro. Image descriptors based on the edge orientation. In *Proceedings of the 4th International Workshop on Semantic Media Adaptation and Personalization*, pages 73–78, Los Alamitos, CA, USA, 2009. IEEE Computer Society. 140
- [184] N. Pinto, D. D. Cox, and J. J. DiCarlo. Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1):151–156, 2008. 62
- [185] A. Pinz, M. Prantl, and H. Ganster. A robust affine matching algorithm using an exponentially decreasing distance function. *Journal of Universal Computer Science*, 1(8):614–631, 1995. 143
- [186] H. Qi, K. Li, Y. Shen, and W. Qu. An effective solution for trademark image retrieval by combining shape description and feature matching. *Pattern Recognition*, 43(6):2017–2027, 2010. 159
- [187] N. J. Radcliffe and P. D. Surry. Fundamental limitations on search algorithms: Evolutionary computing in perspective. In *Lecture Notes in Computer Science 1000*, pages 275–291, Berlin/Heidelberg, 1995. Springer. 61
- [188] V. S. Ramachandran. Filling in gaps in perception: Part I. *Current Directions in Psychological Science*, 1(6):199–205, 1992. 29
- [189] V. S. Ramachandran. Filling in gaps in perception: Part II. scotomas and phantom limbs. *Current Directions in Psychological Science*, 2(2): 56–65, 1993. 30
- [190] Random.org. URL <http://www.random.org/files>. 52
- [191] S. Ravela and R. Manmatha. Multi-modal retrieval of trademark images using global similarity. Technical report, University of Massachusetts, Amherst, MA, USA, 1999. 155
- [192] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Number 15 in Problematika. Frommann-Holzboog, Stuttgart-Bad Cannstatt, 1973. 61
- [193] E. Rosch. On the internal structure of perceptual and semantic categories. In T. E. Moore, editor, *Cognitive Development and the Acquisition of Language*, pages 111–144. Academic Press, New York, 1973. 28
- [194] E. Rosch. Cognitive reference points. *Cognitive Psychology*, 7(4):532–547, 1975. 42, 44, 52
- [195] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439, 1976. 26, 28

Bibliography

- [196] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(6):420–433, 1976. [194](#)
- [197] G. Rote. Computing the Fréchet distance between piecewise smooth curves. *Computational Geometry: Theory and Applications*, 37(3):162–174, 2007. [49](#), [50](#)
- [198] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, New Jersey, 1995. [61](#)
- [199] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. [55](#)
- [200] E. Saund. Finding perceptually closed paths in sketches and drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):475–491, 2003. [110](#)
- [201] L. Scharf. *Probabilistic Matching of Planar Shapes*. PhD thesis, Freie Universität Berlin, 2009. [119](#), [130](#)
- [202] J. Schietse, J. P. Eakins, and R. C. Veltkamp. Practice and challenges in trademark image retrieval. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, pages 518–524, New York, NY, USA, 2007. ACM. [56](#)
- [203] I. Schmitt. *Ähnlichkeitssuche in Multimedia-Datenbanken – Retrieval, Suchalgorithmen und Anfragebehandlung*. Oldenbourg, 2005. [32](#), [33](#), [54](#)
- [204] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, volume 1, pages 755–762. IEEE Computer Society, 2001. [142](#)
- [205] T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125, 2003. [149](#)
- [206] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., London, 1982. [79](#)
- [207] J.-L. Shih and L.-H. Chen. A new system for trademark segmentation and retrieval. *Image and Vision Computing*, 19(13):1011–1018, 2001. [156](#)
- [208] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999. [142](#)

- [209] B. W. Silverman. *Density estimation for statistics and data analysis*. Monographs on statistics and applied probability. Chapman and Hall, London, 1986. [132](#)
- [210] L. Sjöberg. A cognitive theory of similarity. *Göteborg Psychological Reports*, 2(10), 1972. [38](#), [40](#)
- [211] S. S. Stevens. On the psychophysical law. *Psychological Review*, 64: 153–181, 1957. [24](#)
- [212] S. S. Stevens and E. H. Galanter. Ratio scales and category scales for a dozen perceptual continua. *Journal of Experimental Psychology*, 54(6): 377–411, 1957. [24](#)
- [213] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40:361–387, 1987. [119](#)
- [214] A. Sturm. A survey of methods for approximating curves. Technical Report ECG-TR-124101-01, Freie Universität Berlin, 2002. [90](#)
- [215] M. Tanase and R. C. Veltkamp. A straight skeleton approximating the medial axis. In *Proceedings of the European Symposium on Algorithms*, pages 809–821, 2004. [142](#)
- [216] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1974. [139](#)
- [217] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980. [25](#)
- [218] A. M. Treisman and N. G. Kanwisher. Perceiving visually presented objects: recognition, awareness, and modularity. *Current Opinion in Neurobiology*, 8(2):218–226, 1998. [25](#)
- [219] M. Tănase. *Shape decomposition and retrieval*. PhD thesis, Utrecht University, 2005. [14](#)
- [220] M. Tuceryan and A. K. Jain. Texture segmentation using Voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:211–216, 1990. [77](#)
- [221] M. Tuceryan and A. K. Jain. Texture analysis. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *Handbook of pattern recognition and computer vision (2nd Edition)*, pages 207–248. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1998. [77](#)

Bibliography

- [222] A. N. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. [28](#), [38](#), [40](#), [41](#), [42](#), [51](#), [113](#)
- [223] A. N. Tversky and I. Gati. Studies of similarity. In E. Rosch and B. Lloyd, editors, *Cognition and categorization*, pages 79–98. Lawrence Erlbaum Associates, Hillsdale, New York, 1978. [39](#)
- [224] A. N. Tversky and I. Gati. Similarity, separability, and the triangle inequality. *Psychological Review*, 89(2):123–154, 1982. [42](#)
- [225] Y. Umetani and K. Taguchi. Discrimination of general shapes by psychological feature properties. *Digital Systems for Industrial Automation*, 1(2–3):179–196, 1982. [139](#)
- [226] C. Vachier and F. Meyer. The viscous watershed transform. *Journal of Mathematical Imaging and Vision*, 22(2):251–267, 2005. [70](#)
- [227] A. Vailaya, Y. Zong, and A. K. Jain. A hierarchical system for efficient image retrieval. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 3, pages 356–360, Los Alamitos, CA, USA, 1996. IEEE Computer Society. [155](#), [170](#)
- [228] P. Vaxivière and K. Tombre. Celesstin: CAD conversion of mechanical drawings. *Computer*, 25(7):46–54, 1992. [81](#)
- [229] R. C. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht University, the Netherlands, 1999. [35](#)
- [230] R. C. Veltkamp and M. Hagedoorn. Shape similarity measures, properties and constructions. In *VISUAL*, pages 467–476, 2000. [35](#), [36](#), [39](#)
- [231] R. C. Veltkamp and L. J. Latecki. Properties and performance of shape similarity measures. In *Proceedings of the 10th IFCS Conference: Data Science and Classification*, Slovenia, 2006. [19](#)
- [232] R. C. Veltkamp and M. Tănase. Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34 (revised and extended version), Utrecht University, the Netherlands, 2002. [154](#)
- [233] Vienna Classification. International classification of the figurative elements of marks (Vienna Classification) fifth edition. World Intellectual Property Organization, 2002. [57](#)
- [234] Y.-j. Wang and C.-f. Zheng. Trademark image retrieval based on shape and key local color features. In *Proceedings of the International Conference on Information and Computing Science*, volume 2, pages 325–328, Los Alamitos, CA, USA, 2009. IEEE Computer Society. [158](#)

- [235] R. M. Warren. Perceptual restoration of missing speech sounds. *Science*, 167(3917):392–393, 1970. 29
- [236] R. M. Warren, C. J. Obusek, and J. M. Ackroff. Auditory induction: Perceptual synthesis of absent sounds. *Science*, 176(4039):1149–1151, 1972. 29
- [237] E. H. Weber. Der Tastsinn und das Gemeingefühl. In R. Wagner, editor, *Handwörterbuch der Physiologie mit Rücksicht auf physiologische Pathologie. Band 3. Teil 2*, pages 481–588. Vieweg, Braunschweig, 1846. 24
- [238] C.-H. Wei, Y. Li, W.-Y. Chau, and C.-T. Li. Trademark image retrieval using synthetic features for describing global shape and interior structure. *Pattern Recognition*, 42(3):386–394, 2009. 140, 159
- [239] L. Wenyin and D. Dori. A generic integrated line detection algorithm and its object-process specification. *Computer Vision and Image Understanding*, 70(3):420–437, 1998. 81
- [240] Wiktionary. URL <http://en.wiktionary.org>. 13, 20
- [241] H. J. Wolfson. On curve matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:483–489, 1990. 141
- [242] D. H. Wolpert and W. G. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995. 61
- [243] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. 61
- [244] World Intellectual Property Organization. URL <http://www.wipo.int>. 57
- [245] J.-K. Wu, C.-P. Lam, B. M. Mehtre, Y. J. Gao, and A. D. Narasimhalu. Content-based retrieval for trademark registration. *Multimedia Tools and Applications*, 3(3):245–267, 1996. 28, 140, 155
- [246] S.-T. Wu and M. R. G. Márquez. A non-self-intersection Douglas-Peucker algorithm. In *Proceedings of the 16th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'03)*, pages 60–66. IEEE Computer Society, 2003. 91
- [247] X. Wu. An efficient antialiasing technique. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 143–152, New York, NY, USA, 1991. ACM. 19

Bibliography

- [248] S.-S. Xiao and Y.-X. Wu. Rotation-invariant texture analysis using Radon and Fourier transforms. *Journal of Physics: Conference Series*, 48:1459–1464, 2007. [77](#)
- [249] J. Xie, P.-A. Heng, and M. Shah. Shape matching and modeling using skeletal context. *Pattern Recognition*, 41(5):1756–1767, 2008. [149](#)
- [250] P.-Y. Yin and C.-C. Yeh. Content-based retrieval from trademark databases. *Pattern Recognition Letter*, 23(1–3):113–126, 2002. [156](#)
- [251] F. Zernike. Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode. *Physica*, 1(7–12):689–704, 1934. [140](#)
- [252] B.-j. Zou, Y. Yao, and L. Zhang. A new algorithm for trademark image retrieval based on sub-block of polar coordinates. In L. Ma, M. Rauterberg, and R. Nakatsu, editors, *Entertainment Computing – ICEC 2007*, volume 4740 of *Lecture Notes in Computer Science*, pages 91–97. Springer Berlin/Heidelberg, 2007. [158](#)
- [253] S. W. Zucker. Toward a model of texture. *Computer Graphics and Image Processing*, 5(2):190–202, 1976. [77](#)
- [254] S. W. Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5(3):382–399, 1976. [70](#)
- [255] S. W. Zucker and D. Terzopoulos. Finding structure in co-occurrence matrices for texture analysis. *Computer Graphics and Image Processing*, 12(3):286–308, 1980. [77](#)

Summary

THE goal of the present work was to develop a system for automated similarity retrieval of figurative images—especially trademark images—which gives results that resemble human similarity estimation.

In the first chapter, findings about the peculiarities of the perception of images and about human similarity estimation are compiled and the special needs of similarity retrieval of trademark images are explained.

As the depicted shapes play an important role for the estimation of similarity, an approach for the detection of the shapes has been developed. It encounters that shapes may be depicted in different ways (by regions, using textures, by contour lines) and that images often contain compression artefacts and noise.

For the estimation of the similarity of images based on the detected shapes, an approach has been developed that, in a first stage, computes transformations which map the images and, in a second stage, compares the mapped images. For the computation of the transformations an existing randomized approach has been enhanced. It chooses appropriate transformations based on collecting votes. For the comparison of the mapped images a new similarity measure on the contour lines has been developed which takes the correspondences in position and direction into account.

Based on these components a system for similarity retrieval has been developed which also considers the special needs of similarity retrieval of trademark images. The experimental results show a high conformance with human similarity estimation. The results are significantly better than the ones achieved by existing systems.

Zusammenfassung

ZIEL der vorliegenden Arbeit war es, ein System zur automatischen Ähnlichkeitssuche von piktogrammartigen Graphiken, insbesondere von Firmenlogos, zu entwickeln, welches möglichst gute Übereinstimmung mit dem menschlichen Ähnlichkeitsempfinden erzielt.

Im ersten Teil der Arbeit wurden Erkenntnisse über die Besonderheiten der Wahrnehmung von Bildern und über das menschliche Ähnlichkeitsempfinden zusammengetragen sowie die speziellen Anforderungen bei der Ähnlichkeitssuche von Firmenlogos erläutert.

Da die dargestellten Formen die wichtigste Rolle spielen, wurde ein Verfahren für die Detektierung dieser Formen entwickelt. Dabei wurde unter anderem berücksichtigt, dass Formen auf unterschiedliche Art und Weise dargestellt werden können (Flächen, Texturen, Konturlinien) und dass Bilder häufig Fehler wie Kompressionsartefakte und Bildrauschen enthalten.

Zur Ähnlichkeitsbestimmung von Bildern anhand der detektierten Formen wurde ein Verfahren entwickelt, welches im ersten Schritt Transformationen bestimmt, die die Bilder möglichst gut zur Deckung bringen, und im zweiten Schritt die so zur Deckung gebrachten Bilder miteinander vergleicht. Für die Bestimmung der Transformationen wurde ein bestehendes, randomisiertes Verfahren weiterentwickelt, das darauf basiert, anhand von gesammelten Indizien Kandidaten für geeignete Transformationen auszuwählen. Für den Vergleich der zur Deckung gebrachten Bilder wurde ein neues Ähnlichkeitsmaß entwickelt, welches Übereinstimmungen in Position und Richtung der Konturlinien berücksichtigt.

Darauf aufbauend wurde dann ein System zur Ähnlichkeitssuche entwickelt, welches zusätzliche Besonderheiten von Firmenlogos berücksichtigt. Die Ergebnisse der durchgeführten Experimente zeigen eine große Übereinstimmung mit dem menschlichen Ähnlichkeitsempfinden und die erzielten Kennzahlen sind deutlich besser als die, bestehender Systeme.