

# Signature Matching in Document Image Retrieval

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Visual Computing**

eingereicht von

**Thomas Schulz**

Matrikelnummer 0626714

an der

Fakultät für Informatik der Technischen Universität Wien

Betreuung

Betreuer: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig

Mitwirkung: Dipl.-Ing. Dr.techn. Markus Diem

Wien, 25. November 2014

\_\_\_\_\_  
(Unterschrift Verfasser)

\_\_\_\_\_  
(Unterschrift Betreuer)

## Erklärung zur Verfassung der Arbeit

Thomas Schulz  
Buchengasse 41/11  
1100 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 25. November 2014

---

(Unterschrift Verfasser)

I dedicate this work to all my friends and family who kept me company  
throughout my studies.

## Kurzfassung

Das automatisierte Abfragen von Dokumentenbildern anhand bestimmter Kriterien (z. B. nur Dokumente die handgeschriebenen Text enthalten) wird verwendet, um in einer Menge unsortierter Bilder von Dokumenten jene zu finden, die für die vorliegende Aufgabe relevant sind. Um etwa Dokumente zu finden, die von der selben Person verfasst wurden, können händische Unterschriften verwendet werden, sofern die Dokumente Unterschriften enthalten. Daher wird in dieser Arbeit ein System präsentiert, das anhand einer Referenzunterschrift Bilder mit Unterschriften der selben Person findet.

Hierfür wird ein Algorithmus zum Abgleich von Unterschriften mit einem Vorfilterungsschritt kombiniert um festzustellen, welche Unterschriftenbilder zur selben Person gehören. Es werden für jedes Bild jene Thin-Plate Spline (TPS) Transformationen berechnet, die ein Referenzbild am besten auf die anderen Unterschriften abbilden. Aus diesen Transformationen werden anschließend jeweils folgende vier Distanzmaße berechnet: Die Beugungsenergie, die *shape context* Distanz, die anisotrope Skalierung und der Restfehler der Registrierung. Die Gesamtdistanz eines Referenzbilds zu den anderen Unterschriften wird berechnet, indem die vier Distanzmaße mit Hilfe von Gewichten kombiniert werden, welche über Lineare Diskriminanzanalyse (LDA) ermittelt werden. Mit Hilfe eines Vorfilterungsschritts, der auf der *shape context* Distanz basiert, wird der Vorgang beschleunigt, so dass die zeitaufwendige Berechnung der Transformationen und der Distanzmaße nur mehr für 3% der Unterschriften durchgeführt werden muss.

Für die Evaluierung werden zwei Teilmengen der *GPDS960signature* Datenbank verwendet. Der hybride Ansatz dieser Arbeit erreicht eine um 1,6 Prozentpunkte höhere MRP und eine um 0,9 Prozentpunkte höhere MAP als die Distanzmaße allein, während gleichzeitig die Laufzeit auf der größeren Teilmenge um das 16-fache verringert werden kann. Die Ergebnisse zeigen außerdem, dass die gewichtete Implementierung des Restfehlers der Registrierung, welche in dieser Arbeit vorgeschlagen wird, eine deutlich höhere MAP erreicht (54,8% statt 7,6%), als eine auf exakten Korrespondenzen basierende Implementierung. Weiters wird gezeigt, dass das vorliegende System zum Abgleich von Unterschriftenbildern nicht auf Trainingsdaten angewiesen ist, da die Ergebnisse mit 25% Trainingsdaten höchstens 0,3 Prozentpunkte besser sind als ohne Trainingsdaten.

## Abstract

Document image retrieval is a method used for searching through unsorted images of documents to find the ones which are relevant for a given task. One way to do this is by sorting the documents according to their corresponding author. For documents that contain handwritten signatures this can be done using signature matching techniques. This thesis therefore presents a signature retrieval system that uses query signature images in order to find images with signatures from a specific author.

The retrieval system uses a state-of-the-art signature matching algorithm that is combined with a pre-filtering step to determine which signature image belongs to the same author. For this purpose it computes the Thin-Plate Spline (TPS) transformation that best maps one signature image to another for each signature image in a library. This transformation is used to compute four dissimilarity measures, namely the bending energy, the shape context distance, the anisotropic scaling and the registration residual error. The total dissimilarity of the query signature to the other signatures is computed by combining the four dissimilarity measures using weights obtained by means of Linear Discriminant Analysis (LDA). The whole process is sped-up by applying a pre-filtering step based on the shape context distance to the query signature such that the time-consuming computation of the transformation and the dissimilarity measures is done for only 3% of the images in the library.

The evaluation of the signature retrieval system is done using two subsets of the *GPDS960signature* database. The hybrid approach proposed in this thesis achieves a 1.6 percentage points higher MRP and a 0.9 percentage points higher MAP than the dissimilarity measures on their own, while providing a speed-up of factor 16 on the larger set. The results also demonstrate that the weighted implementation of the registration residual error that is proposed yields a significantly higher MAP (i.e. 54.8% compared to 7.6%) than the implementation which uses exact matches. Furthermore, it is shown that the retrieval system is not dependent on the use of training data as the performance with 25% training data is at most 0.3 percentage points better than without any training data.

## Acknowledgements

I want to thank Miguel A. Ferrer for providing the *GPDS960signature* database that is used for the evaluation in this thesis. I would also like to thank Robert Sablatnig and Markus Diem for their valuable input and support.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.1.1	Scope of Discussion . . . . .	3
1.1.2	Objective . . . . .	3
1.1.3	Contribution . . . . .	3
1.2	Definition of Terms . . . . .	4
1.3	Results . . . . .	6
1.4	Thesis Structure . . . . .	7
<b>2</b>	<b>Related Work</b>	<b>8</b>
2.1	Layout Analysis . . . . .	8
2.1.1	Handwritten Text Localization . . . . .	9
2.1.2	Signature Localization . . . . .	10
2.2	Preprocessing . . . . .	12
2.2.1	Binarization . . . . .	12
2.2.2	Noise Removal . . . . .	12
2.2.3	Printed Text Removal . . . . .	13
2.2.4	Normalization . . . . .	14
2.2.5	Abstraction . . . . .	14
2.3	Signature Matching . . . . .	15
2.3.1	Feature Extraction . . . . .	15
2.3.2	Verification and Identification . . . . .	20
2.3.3	Retrieval . . . . .	22
<b>3</b>	<b>Methodology</b>	<b>26</b>
3.1	Theoretical Background . . . . .	27
3.1.1	Thin-Plate Splines . . . . .	27
3.1.2	Shape Context Descriptor . . . . .	29
3.1.3	Linear Discriminant Analysis . . . . .	31
3.2	Preprocessing and Point Sampling . . . . .	33
3.3	Thin-Plate Spline – Robust Point Matching Algorithm . . . . .	36

3.4	Dissimilarity Measures . . . . .	38
3.4.1	Bending Energy . . . . .	39
3.4.2	Shape Context . . . . .	40
3.4.3	Anisotropic Scaling . . . . .	40
3.4.4	Registration Residual Error . . . . .	40
3.5	Pre-Filtering . . . . .	41
<b>4</b>	<b>Results</b>	<b>43</b>
4.1	Registration Residual Error . . . . .	45
4.2	Orientation Normalization Techniques . . . . .	46
4.3	Abstraction Techniques . . . . .	47
4.4	Sampling Techniques . . . . .	50
4.5	Hybrid Approach . . . . .	53
4.5.1	Reduced Set Size . . . . .	54
4.5.2	Comparison with Zhu et al. . . . .	55
4.5.3	Training Data . . . . .	56
4.5.4	Single Distances . . . . .	57
4.6	Prototype . . . . .	59
<b>5</b>	<b>Conclusion</b>	<b>63</b>
5.1	Disadvantages . . . . .	64
5.2	Advantages . . . . .	64
5.3	Future Work . . . . .	65
	<b>Bibliography</b>	<b>66</b>
	<b>List of Acronyms</b>	<b>72</b>



# Introduction

In order to analyse libraries of unsorted documents it is helpful to be able to automatically find documents which meet certain criteria (e.g. only documents with handwritten text). One example of such a library are the so-called Stasi-files. The *Ministry for State Security* (Ministerium für **Staatssicherheit**, briefly Stasi) was founded in 1950 as the secret service of the German Democratic Republic (GDR). Over the years it collected records on the citizens of the GDR in the form of documents, photographs, videos and audiotapes. Before its dissolution in 1990, members of the Stasi started to destroy a portion of the records. Since the electronic paper shredders could not handle the amount of documents, a large part was torn by hand [47]. The results of this are more than 16,000 bags containing over 600 million hand-torn paper fragments. Since manual reconstruction is estimated to take 30 people 600 to 800 years [47], members of the Fraunhofer Institute for Production Systems and Design Technology (IPK)<sup>1</sup>, Berlin and the Computer Vision Lab (CVL)<sup>2</sup> at Vienna University of Technology are working on a system for automated reconstruction. Besides putting torn pieces together to documents, the system also has to group documents together which belong to the same record.

## 1.1 Motivation

One way of sorting libraries of documents is by determining the corresponding author of each document. In context of the Stasi files this additional information can be used to group documents together which belong to the same record. Since documents that do not contain any handwritten text cannot be assigned to an

---

<sup>1</sup><http://www.ipk.fraunhofer.de>

<sup>2</sup><http://www.caa.tuwien.ac.at/cvl/>

author using handwriting recognition, other means have to be considered in order to determine the author. An effective means for doing this is the use of signature matching techniques [1, 61] provided that the document contains a signature.

There is a distinction between offline and online signature matching, where online means that the signature is captured using an electronic device which also captures temporal information on the stroke sequence. Thus the signature can be parameterised over time and the matching can be done using methods like Dynamic Time Warping (DTW) [1]. In offline signature matching, on the other hand, no electronic device is needed to record the stroke sequence, however, only static information is available for matching and there is no inherent 1-D parametrisation of the signature [1].

Signature matching is used in areas such as verification [57], identification [49] and retrieval [48]. While signature verification deals with confirming the genuineness of a signature and signature identification tries to find the corresponding author [40], signature retrieval aims at finding document images that contain signatures from a specific individual [48]. The differences between the three categories are illustrated in Figure 1.1. It shows the respective problems that have to be solved for signature verification (left), identification (middle) and retrieval (right).

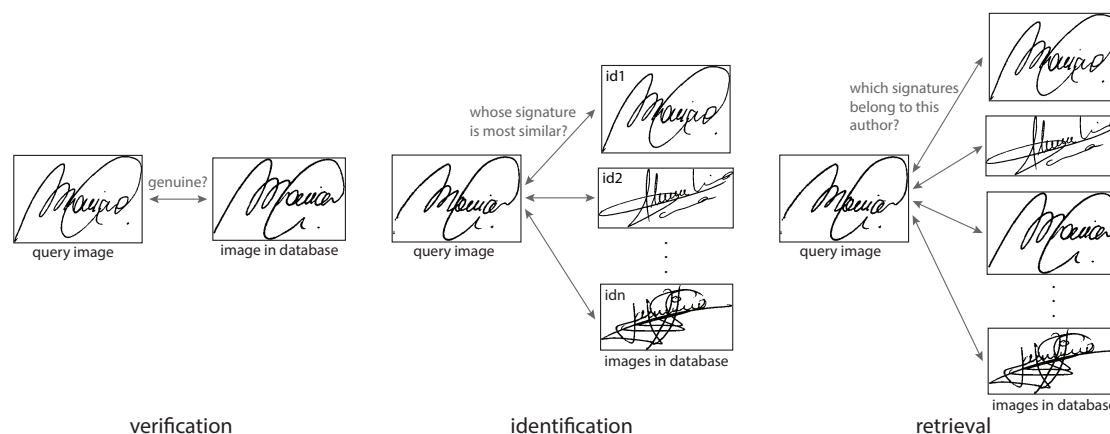


Figure 1.1: An illustration of the differences between the three areas of application for signature matching. Figure inspired by [40].

In contrast to signature verification, where both online and offline recognition methods are applicable [27], document image retrieval deals only with offline methods. Another difference between signature verification and document image retrieval using signatures is that verification is a one-to-one problem (i.e. a test signature is compared to a stored reference signature to verify the identity of the signer) whereas document image retrieval is a one-to-many problem (i.e. a test

signature is compared to all other signatures in the dataset to find the documents with matching signatures).

### 1.1.1 Scope of Discussion

This thesis focuses on the matching aspects of a signature-based document image retrieval system. The aim is to analyse state-of-the-art signature retrieval methods and enhance them by applying a pre-filtering step. The scientific question can therefore be posed as: *Can the runtime of a state-of-the-art signature retrieval system be reduced by means of a pre-filtering step without decreasing the retrieval performance?*

Since the data set used for evaluating the signature retrieval system in this thesis contains binary images of segmented signatures, there is no binarization included in the preprocessing of the presented system. Furthermore, there is also no printed text removal implemented in the retrieval system which would be necessary for real-world documents since the signature can overlap with the person's machine-printed name or the complementary close.

A complete document image retrieval system would also require the localization of the signature in the document and its segmentation. The retrieval system in this thesis, however, only deals with the matching and retrieval part and is therefore more accurately called a signature retrieval system.

### 1.1.2 Objective

The task of signature matching and retrieval comprises several challenges. Firstly, in contrast to handwriting, signatures are more stylistic and not necessarily related to a sequence of characters [61]. Existing approaches towards document retrieval using handwriting can therefore not be used for signatures. Secondly, due to the stylistic nature of signatures, different signatures from the same person appear differently. This can be seen in Figure 1.2 where each row shows different signatures of one author. Hence, the signature matching algorithm of the retrieval system in this thesis has to be flexible enough to recognise different signatures from the same author as being similar, while still separating them from the signatures of other individuals.

### 1.1.3 Contribution

The aim of this thesis is to analyse state-of-the-art signature retrieval approaches and develop software that is able to scan libraries of signature images using a query signature. It provides an introduction to the topic and gives an overview of related areas. The retrieval system based on the approach of Zhu et al. [61]

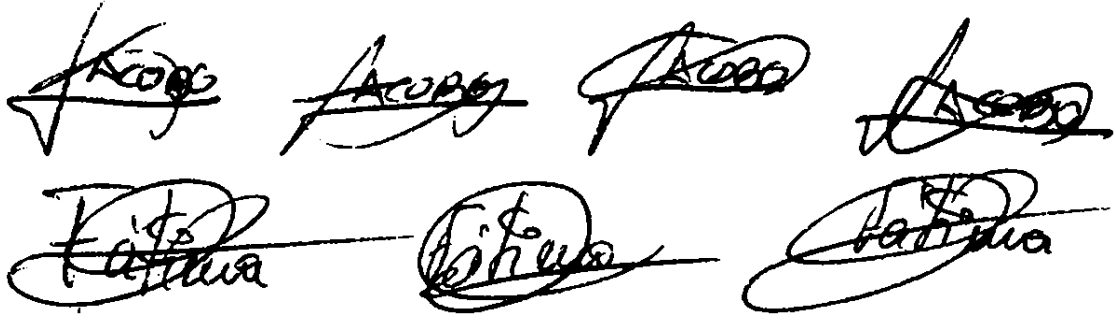


Figure 1.2: Each row depicts different signatures of one author that show great variation in their appearance.

(see Chapter 3) is tested using different abstraction techniques and point sampling rates to evaluate the effects on the retrieval performance.

Furthermore, a weighted implementation of one of the dissimilarity measures used by Zhu et al. (i.e. the registration residual error) is proposed and it is shown that it yields a significant increase in retrieval performance compared to the implementation that uses exact matches.

The main contribution of this thesis is the suggestion to combine the approach of Zhu et al. with a shape-context-based pre-filtering step to reduce the runtime. Due to the computational load of the approach and the fact that the dissimilarity measures have to be computed for the entire test set for each new query signature, since they depend on the transformations between the query signature and the candidate signatures, the approach becomes infeasible for larger datasets. The runtime reduction provided by the hybrid approach proposed in this thesis therefore extends the retrieval system such that it can be used for larger sets of signature images.

## 1.2 Definition of Terms

This section gives an overview of terms and abbreviations that are used throughout this thesis.

**BFE Best-Fit Ellipse:** The ellipse which best fits a two-dimensional object. It can be used to normalize the orientation of the object by aligning the ellipse's major or minor axis with the horizontal axis (see Section 3.2).

- CC Connected Component:** In visual computing, foreground pixels that are connected with each other form a so-called CC. It is distinguished between 4-connectivity where only horizontal and vertical connections are allowed and 8-connectivity where diagonal connections are allowed as well.
- EMD Earth Mover's Distance:** A distance metric that is used to compute the difference between two histograms. In contrast to the simpler  $\chi^2$  test statistic it also considers cross-bin relationships [45] (see Section 2.3.3).
- HMM Hidden Markov Model:** A statistical model used in pattern recognition where each HMM represents a pattern that can be recognized. The model with the highest probability of producing a given feature vector determines the outcome of the recognition process [43] (see Section 2.3.2).
- ICP Iterative Closest Point:** A fast iterative method for registering point clouds using nearest-neighbour relationships [3].
- LDA Linear Discriminant Analysis:** A means of finding directions that efficiently separate different classes of data based on the Fisher classifier [19]. It can also be used to combine several features by taking the coefficients of the separating hyperplane as weights (see Section 3.1.3).
- MAP Mean Average Precision:** A performance measure in document image retrieval that rewards higher rankings of relevant documents while penalizing that of irrelevant ones [6, 61] (see Chapter 4).
- MRP Mean R-Precision:** A performance measure in document image retrieval that is computed as the precision at a certain rank which means that it ignores the exact ranking of the documents [6, 61] (see Chapter 4).
- NN Neural Network:** A statistical model with the aim to emulate the human brain. It consists of several layers with interconnected neurons and has to be trained before it can be used for classification [55] (see Section 2.3.2).

- SIFT **Scale-Invariant Feature Transform:** A local descriptor that is invariant to changes in image scale and rotation. Scale invariance is achieved using difference-of-Gaussian images and rotation invariance is accomplished using a histogram of gradient directions and assigning each descriptor one or more main directions [34].
- SVM **Support Vector Machine:** A machine learning technique that can be used to classify data using a hyperplane. This hyperplane is obtained by training the SVM such that it separates the two classes with maximum margin [11, 37] (see Section 2.3.2).
- TPS **Thin-Plate Spline:** A model that can be used to describe a transformation with an infinite thin metal plate that is deformed according to the transformation. It is able to model affine and non-rigid transformations such that they can be separated [5] (see Section 3.1.1).
- TPS-RPM **Thin-Plate Spline – Robust Point Matching:** A method for registering point clouds using a TPS to model the transformation from one point cloud to another [10] (see Section 3.3).

## 1.3 Results

The retrieval system proposed in this thesis is evaluated on two subsets of the *GPDS960signature* database [4] which contains 24 binary signature images each of 960 individuals. The performance is measured in terms of Mean Average Precision (MAP), Mean R-Precision (MRP) and runtime since the former two are the most common measures in document image retrieval [61] and the latter is an important factor of the retrieval system in discussion (see Section 3.5).

The evaluation shows that the hybrid approach proposed in this thesis is able to achieve a 1.6 percentage points higher MRP and a 0.9 percentage points higher MAP than the dissimilarity measures on their own, while providing a speed-up of factor 16. It is also shown that increasing the reduced set size yields further performance gain of up to 1.9 percentage points while still providing a speed-up of factor 13.

The results also show that the signature retrieval system does not require the use of training data. The hybrid approach achieves an only 0.1 percentage points higher MAP with 25% training data than without any training data. The dissimilarity measures on their own even perform best when not using training data at all.

Furthermore, the evaluation demonstrates the performance gain obtained from the two implementations of the registration residual error that are proposed in this thesis (see Section 3.4.4). More precisely, the weighted registration residual error (i.e.  $D_{re}^W$ ) achieves an MAP of 54.8% at a runtime of 15.6 hours on the smaller test set while the implementation that uses exact matches (i.e.  $D_{re}^H$ ) achieves merely an MAP of 7.6% at a runtime of 25.4 hours.

## 1.4 Thesis Structure

The outline of this thesis is as follows. The next chapter discusses state-of-the-art methods related to the subject of document image retrieval. It starts by giving a short introduction to layout analysis while focusing on handwritten text and signature localization. It subsequently presents common preprocessing methods and describes the state of the art of signature matching techniques and their respective fields of application.

Chapter 3 describes all parts of the signature retrieval system presented in this thesis. First, it gives an introduction to the techniques used in the matching algorithm and explains the workflow of the retrieval system. It then describes the preprocessing and point sampling techniques that are used and discusses the algorithm for matching two signature images. The chapter is concluded by presenting the four dissimilarity measures used for signature retrieval along with the pre-filtering step that is suggested in this thesis.

The various parts of the retrieval system are evaluated in Chapter 4. First, three different implementations of one of the dissimilarity measures (i.e. the registration residual error) and three orientation normalization techniques are evaluated. The following tests evaluate the effects of different abstraction methods (i.e. edge detection and skeletonization) and varying sampling rates. The hybrid approach presented in this thesis is then evaluated and compared to the performance of the dissimilarity measures on their own.

Finally, Chapter 5 concludes the thesis by discussing advantages and disadvantages of the retrieval system presented in this thesis and also gives suggestions for future work on the topic.

## Related Work

This chapter is intended to give an overview of all aspects involved in a document image retrieval system that is based on handwritten signatures. The first part provides a general introduction to layout analysis and the task of finding a signature in a document. Section 2.1.1 deals with the more general case of handwritten text localization while Section 2.1.2 discusses signature localization in particular. The second part addresses the subject of preparing a signature image for further processing by means of common preprocessing techniques and the third part focuses on signature matching in different areas of application.

The third part of this chapter is split into three sections. Section 2.3.1 gives an overview of feature extraction techniques that are used for signature matching and Sections 2.3.2 and 2.3.3 present some methods from the fields of signature verification, identification and retrieval respectively.

### 2.1 Layout Analysis

For completely automated analysis of document images, it is first necessary to analyse their layout [20]. This allows to classify the image according to its content and segment certain areas such as machine-printed text or handwritten signatures. Further algorithms can then be applied to a specific region only instead of the entire document image.

Document segmentation approaches are generally split into hierarchical and non-hierarchical methods [52]. Hierarchical methods work either in a top-down, bottom-up or combined manner to segment different regions, while non-hierarchical methods ignore the geometric relation among blocks [52]. Top-down approaches split the document into coarse regions and then recursively segment each region



into finer ones. It is distinguished between basic objects, which are the leaf nodes of the segmentation tree, and composite objects which are all other nodes [52]. Bottom-up approaches on the other hand start from Connected Components (CCs) and merge them into successively larger regions based on local features. There are also hybrid approaches (e.g. [36]) which use a combined top-down and bottom-up analysis for segmenting document images.

Top-down approaches are faster than bottom-up approaches but primarily effective for documents with a known layout (e.g. forms) [52]. Bottom-up approaches are more flexible than top-down approaches regarding the document layout but are also more computationally expensive [52]. This is why top-down approaches are considered as being *knowledge driven* while bottom-up approaches are so-called *data driven* methods [52].

Text localization techniques are split into region-based and texture-based methods [24]. Region-based approaches use CCs or edges and merge them into Bounding Boxes (BBs) of text regions in a bottom-up manner. Texture-based approaches on the other hand use methods like Gabor filters, wavelets or Fast Fourier Transform (FFT) to detect distinct textural features of text areas [24].

Layout analysis on documents that contain noise (e.g. vertical and horizontal lines, dots and elements which are particularly small compared to the remaining elements in the document) can be improved by applying noise removal before classifying the elements of a document [44]. Other preprocessing methods in layout analysis include binarization of the document and skew estimation [13].

Jung et al. [23] find CCs using Canny edge detection [9] followed by dilation and erosion operations. The CCs are refined using their horizontal and vertical projection profiles. Each line is then classified as text or non-text using a Support Vector Machine (SVM). Finally the text BBs are refined using the SVM output scores. Similarly, Messaoud et al. [35] do Canny edge detection and CC merging based on BB intersections.

The following sections focus on localization methods which are specifically used for handwritten text and signatures respectively.

### 2.1.1 Handwritten Text Localization

In order to classify segments as handwritten text it is necessary to extract features which can discriminate handwritten text from other document elements such as machine-printed text. An early approach by Kuhnke et al. [31] uses the straightness of vertical and horizontal lines together with the symmetry relative to the centre of gravity of each character. This is based on the idea that Roman characters consist mostly of straight lines, while characters without straight lines are described by the symmetry feature. The discrimination between handwritten and machine-printed text is done using a feed-forward Neural Network (NN) with three active layers.

This method, however, only works on single images of handwritten and printed characters where it achieves a recognition rate of 78.5%.

An approach that localizes and discriminates handwritten text from other document elements is presented by Kavallieratou and Balcan [28]. They use eight low-level features of first-order and second-order CCs, including the black pixel density, three symmetry features and the ratio of distinct heights in second-order CCs. Decision rules based on thresholds are used to decide whether a first-order CC is classified as machine-printed or handwritten. The decisions are based on observations such as the smaller aspect ratio and higher black pixel density of machine-printed text compared to handwritten text [28]. This approach has a high recognition rate of 96% on first-order CCs, however, it relies on the correct selection of five thresholds which is difficult for general data.

A different approach is presented by Diem et al. [13] who propose a Gradient Shape Feature (GSF) which is based on the shape context feature of Belongie et al. [2]. The GSF is computed for a square sliding window that is moved along the principal axis of the BB of a CC. For each sliding window an SVM with a Radial Basis Function (RBF) kernel is used to determine the class (i.e. handwritten, machine-printed or noise) and a weight which corresponds to the log likelihood of the SVM for the feature. The final class of the CC is computed by accumulating the weights of all corresponding sliding windows. The classification is further improved by using back propagation to correct the class labels of falsely classified CCs. This method does not rely on the manual selection of thresholds, but is based on machine learning methods and achieves a precision of 92.4% on real-world data.

### 2.1.2 Signature Localization

Esteban et al. [16] state that automatic localization of signatures in documents received little attention in literature which is why it is still an open problem [61]. One reason for this is that in the area of signature verification on cheques the approximate location of the signature can be found through prior knowledge of the document structure (see Figure 2.1). An overview of methods for form document processing is given in [52].

Ramakrishnan et al. [44] propose to use Zernike moments [53] as features for classification of CCs with an SVM having an RBF kernel. To refine the results they perform Delaunay triangulation [12] on the CCs and use the attributes of neighbouring triangles to reassign class labels. If more than 50% of the neighbouring triangles have similar features but a different class label, their label is assigned to the current triangle. However, Ramakrishnan et al. do not differentiate between handwritten text and signatures but merely classify both as handwritten text.

An approach that focuses particularly on the localization of signatures on cheques using evidence accumulation is proposed by Esteban et al. [16]. They make

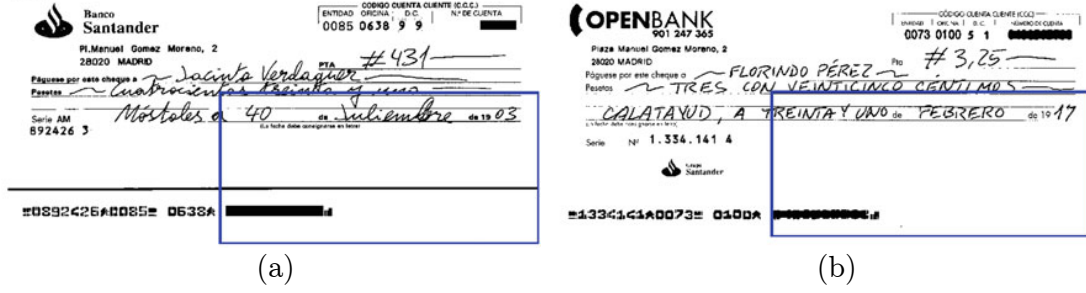


Figure 2.1: Two cheques with the signature search region marked in blue. Image taken from [16].

use of the fact that in signature verification a reference signature is known beforehand. The contours of this signature and the signature search region (see Figure 2.1 where the signature search region is marked by a blue rectangle), which on a cheque corresponds approximately to the bottom right corner [16], are extracted through a morphological operator. Each contour is split into smaller segments which are described by a 4-tuple  $(x, y, \Theta, \kappa)$ , where  $x$  and  $y$  are the coordinates of the middle point of the segment,  $\Theta$  is the angle of its tangent through that point and  $\kappa$  is the curvature of the segment. Subsequently a correspondence vector is computed by subtracting the signature tuple from the search region tuple and normalizing  $\Theta$  to  $[-\pi/2, \pi/2]$  for each possible combination of segments from the signature and the search region. After a clustering step of the correspondence vectors the larger cluster determines the location of the signature in the search region. This step is repeated after reducing the limits of the grid used for clustering towards the largest cluster until a fixed amount of time has passed. This approach, however, is only applicable when the signature is known in advance.

Zhu et al. [61], on the other hand, propose a method that localizes signatures in arbitrary documents without making any assumptions on the appearance of the signature. They present a multiscale detection algorithm that uses curvature properties to compute saliency values based on a signature production model that interprets signatures as concatenations of elliptic segments. Firstly the document image is smoothed with a Gaussian kernel and resampled using a Lanczos filter [7] before edges are extracted using a Canny edge detector. By applying this procedure to the document image at multiple scales small gaps in curves are filled, thus reconnecting contours that are broken due to poor ink condition or image degradation. Secondly, CCs are extracted from the edge image and saliency values are computed for each of them. The final result of the localization is obtained by connecting the most salient CCs with neighbouring CCs based on curvilinear constraints. The localization results are further improved by using document context,

i.e. only curves at the bottom of the document are used for signature detection.

## 2.2 Preprocessing

Before a located signature can be used in a matching algorithm it is necessary or simply advantageous to apply further preprocessing methods on the image. Depending on the localization and matching techniques that are used, some or all of the following methods can be applied.

### 2.2.1 Binarization

Binarization is commonly used as a preprocessing step in the area of signature verification [26, 39] and identification [21, 40]. An early binarization approach is presented by Otsu [38] who computes a global threshold for an image that separates foreground and background based on the histogram of its intensity values. However, global thresholding techniques like this one fail for documents which are not separable into foreground and background by a single threshold (e.g. due to changing illumination) [14].

An adaptive binarization method is proposed by Sauvola and Pietikäinen [46] who use a so-called *hybrid switch* that quickly differentiates between text regions and non-text regions. The text regions are then binarized by computing a threshold for each pixel using the local mean, standard deviation and the dynamic range of the standard deviation in a neighbourhood window. They reduce the computation time by computing the threshold for every  $n$ th pixel only and interpolating the values for the remaining pixels. The computation can be further sped up using integral images, thus making the computation complexity independent of the neighbourhood window size [60].

A more recent approach by Kleber et al. [29] uses a scale space for binarization, thus making the algorithm independent of script size and allowing to use the coarser scales as a foreground estimation. In addition, the Gaussian smoothing of the coarser scales removes high frequency noise which makes the binarization more robust. The image at each scale is binarized using the local maximum and minimum approach of Su et al. [51].

### 2.2.2 Noise Removal

A widely used method [26, 39, 48] for reducing the noise in an image is to apply a median filter. This filter assigns each pixel the median intensity value of the pixels in its proximity. It is effective against salt and pepper noise (i.e. light and dark dots on the image) while preserving edges (in contrast to a Gaussian filter for

example) [39]. Figure 2.2 illustrates this by adding salt and pepper noise to the signature image in (a) and filtering the resulting image in (b) using a median filter (c) and a Gaussian filter (d). The results in (c) and (d) show that the median filter is able to remove most of the noise while preserving the signature edges whereas the Gaussian filter blurs the image and is unable to remove the noise. Other possibilities include the removal of noise based on custom criteria (e.g. size) or using morphological operators. The latter can also be used to fill small gaps in broken contours which can occur due to poor ink condition.

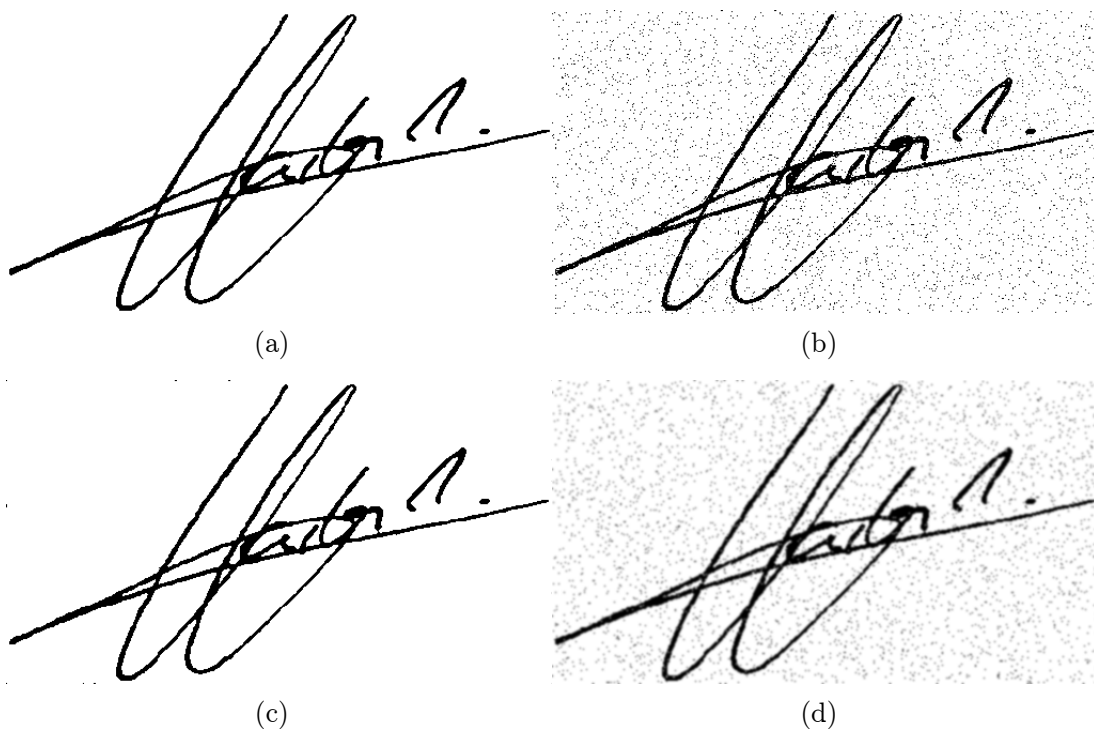


Figure 2.2: (a) The original signature image. (b) The image from (a) with 5% salt and pepper noise. (c) The result of filtering (b) with a  $3 \times 3$  median filter. (d) The result of filtering (b) with a  $5 \times 5$  Gaussian filter with  $\sigma = 0.83$ .

### 2.2.3 Printed Text Removal

For document images where the signature overlaps with machine-printed text it is necessary to remove that text before using the signature for further processing. Srihari et al. [50] do so by using the aspect ratio of the BB of a CC and the ratio of its size compared to the largest CC in the image as features for a Fisher linear discriminant classifier. However, CCs which are labelled as printed text are only

removed if they are in a line with other CCs that are labelled as printed text. The last step is to find and remove isolated CCs by analysing the neighbourhood of the remaining CCs that are classified as machine-printed text.

### 2.2.4 Normalization

In case the features which are used for matching are not size and rotation invariant, it is necessary to normalize the signature images accordingly. Pavlidis et al. [40] for example perform orientation normalization by aligning the elongation axis of each signature with the horizontal axis, but do not normalize the size of the signature images because they use size invariant features. Since their orientation normalization approach has two possible outcomes for each signature image (depending on whether the image is rotated to the left or to the right), they use both results for their signature identification algorithm.

Oz [39] and Karki et al. [26] normalize the signature images to a fixed size, thereby changing the aspect ratio between width and height of the image. On the other hand, Lin and Chang [33] preserve aspect ratios by normalizing the image size according to the length of its diagonal. They also perform orientation normalization similar to Pavlidis et al. [40] by computing the Best-Fit Ellipse (BFE) of the signature and aligning its major axis with the horizontal axis.

### 2.2.5 Abstraction

By computing an abstract representation of the signature it is possible to reduce computational complexity and to aid shape comparison [61], e.g. by reducing the image to curve segments. One method to do this is to compute the skeleton of the image by means of a thinning or skeletonization algorithm as the one proposed by Zhang and Suen [59]. This method has the advantage of removing stroke thickness information, thus making a subsequent matching algorithm independent of the pen that is used for signing. However, thinning also introduces artefacts such as spurs [42]. A Canny edge detector on the other hand provides a more robust abstraction [61], but also captures the stroke thickness since each stroke results in two edges. The differences of these abstraction techniques are visualized in Figure 2.3 where the second row shows augmentations of the areas marked in the first row. It depicts a binary signature image on the left, the skeleton of this signature in the middle and its edges on the right.

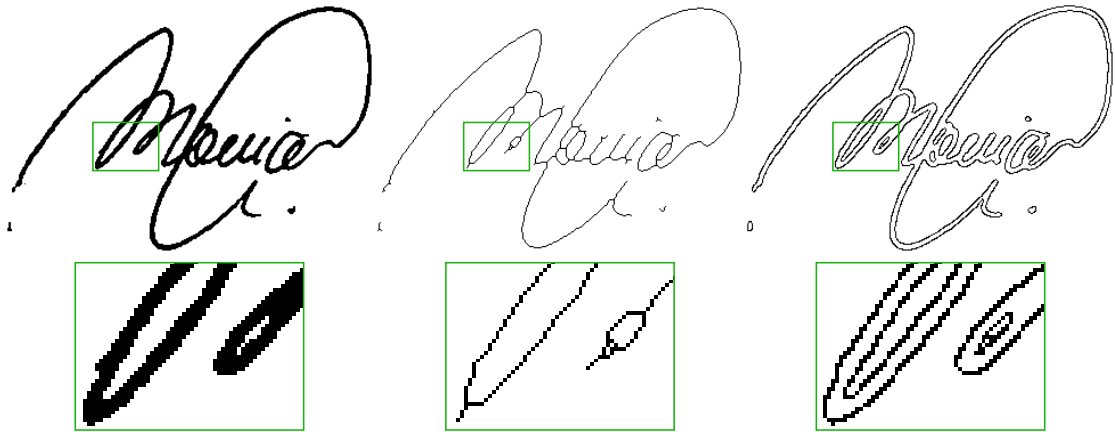


Figure 2.3: The first column shows the original signature image, the second column shows the result of a thinning algorithm and the last column shows the Canny edges. The second row shows augmentations of the areas which are marked in the first row.

## 2.3 Signature Matching

The matching stage consists of two steps, namely the extraction of features from the signature image and the comparison of these features. Depending on the application the features are used to verify that the signature belongs to a specific individual (signature verification), to find the corresponding author (signature identification) or to retrieve the signatures from the same author in a dataset (signature retrieval). The following sections present different feature extraction techniques and discuss how they are used in signature verification, identification and retrieval.

### 2.3.1 Feature Extraction

Two early approaches to signature matching by Han and Sethi [21] and Pavlidis et al. [40] are both based on string representations. Han and Sethi first detect vertical and horizontal bars, loops, end points, branch points, crossing points, convex points and concave points in each signature image. They subsequently project the features onto the horizontal and vertical axis of the image and construct two strings which contain the features for each axis in left-to-right and top-to-bottom order respectively. Each feature is replaced by a letter representation (e.g. L for loop, B for branch point, etc.) for this purpose. This allows the use of partial queries where parts of the signature are missing. However, it can easily happen that features change their position in the string representation of different

signatures from the same author. Figure 2.4 illustrates this behaviour on a simple example with edge (purple) and branch (green) points on two signatures from the same person. It shows that the horizontal and vertical order changes significantly even when only these two features are detected.

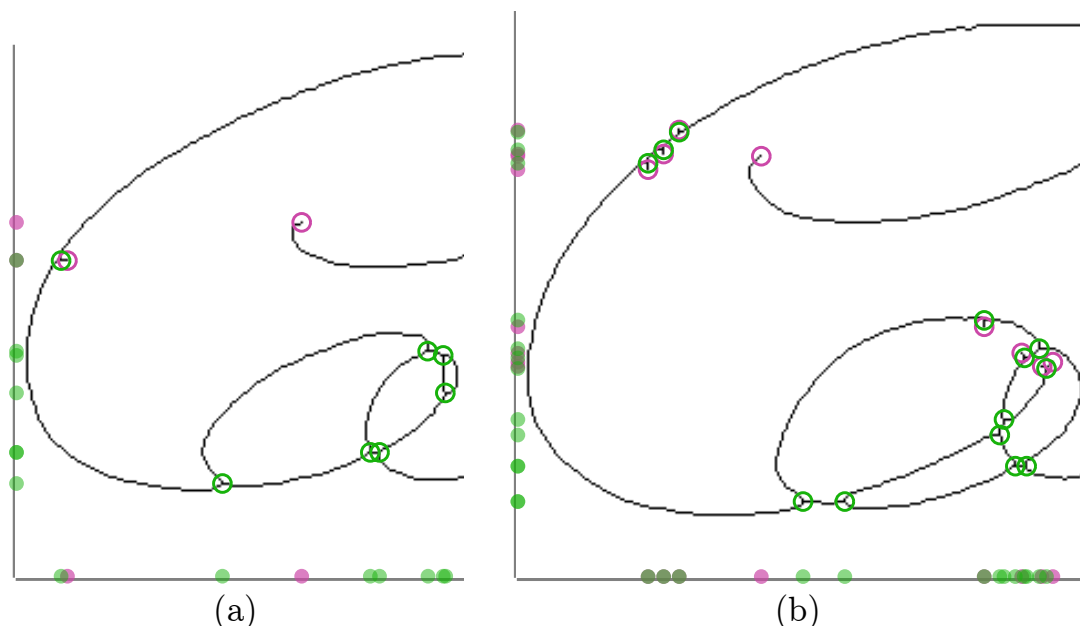


Figure 2.4: Parts of two different signature images from one person with edge and branch points marked in purple and green respectively. The dots on the axes indicate the horizontal and vertical order of the detected points.

Pavlidis et al. extract a more robust string representation at the cost of higher computational complexity. This is done by first placing two opposing particles on the enclosing ellipse of the signature which are connected by an elastic spring through the centre of the signature. The particles are attracted by the force of the spring and a virtual gravity field around the signature until they rest on the contour of the signature. This step is repeated for two new particles which are placed 5 degrees rotated from the last position until all 72 (i.e.  $360/5$ ) particles are lying on the signature. The sequence of these particles is then used as an abstract representation of the signature shape. The noise in this abstraction is subsequently reduced by computing a polygonal approximation which smooths small fluctuations of almost straight lines. This polygon is eventually transcribed into a string representation by following its edges and storing the angle  $\theta$  between two adjacent edges as a letter  $A, \dots, R$  (i.e.  $A$  for  $\theta \in (0^\circ, 20^\circ]$ ,  $B$  for  $\theta \in (20^\circ, 40^\circ]$ , etc.).



The following three approaches use a grid-segmentation scheme to extract local features. The first is presented by Justino et al. [25] who compute two static and two pseudo-dynamic features for each grid cell, namely: The pixel density as the number of black pixels divided by the total number of pixels per cell, the gravity centre distance, the stroke curvature as the curvature angle of the biggest stroke in the cell and the predominant slant. Each column of the grid is then converted to a feature vector for each of the four features. These vectors are subsequently converted to a codebook using a vector quantization procedure with a  $k$ -means algorithm.

A less complex method in terms of that it does not require the generation of a codebook is presented by Srihari et al. [50]. They compute a binary feature vector consisting of Gradient, Structural and Concavity (GSC) features [17] which are extracted for each cell of a  $4 \times 8$  grid. To compute the gradient feature vector, the binary image is first convolved with two Sobel operators to approximate the  $x$  and  $y$  derivatives of the image (see Figure 2.5 (a) and (b)). The derivatives are then combined using vector addition to get the image gradient (see Figure 2.5 (c) and (d)). For each cell a histogram is computed from the gradient directions and thresholded to get the binary gradient feature vector. The structural feature vector is computed by passing  $3 \times 3$  operators over each cell of the gradient image to detect patterns and combining them to larger features using a rule table. The concavity feature vector is composed of 8 feature values for each grid cell. One value is computed by counting the number of foreground pixels and thresholding it. Two values are used to indicate the presence of longer horizontal or vertical lines by counting the run lengths of signature pixels and applying a threshold. The remaining five values are computed using a *star-operator* which shoots rays from each pixel in 8 directions. Depending on which ray hits a foreground pixel and which ray hits the image border, this operator yields different patterns which are used to classify the region as upward, downward, left or right concavity or as a hole. In combination this results in a 1024 bit binary feature vector. The quality of this feature vector, however, relies heavily on the selection of the thresholds that are used for the binarization of the feature values.

The approach of Karki et al. [26] neither requires the generation of a codebook nor the assumption of thresholds. They use a simple grid feature together with several global features. The grid feature is computed by splitting the image into 96 ( $12 \times 8$ ) segments and counting the number of foreground pixels in each segment. The values are normalized such that the lowest value becomes zero and the highest value becomes one. The global features that are used are:

- Image area – the number of foreground pixels in the image.
- Pure width and height – the width and height of the image after all rows

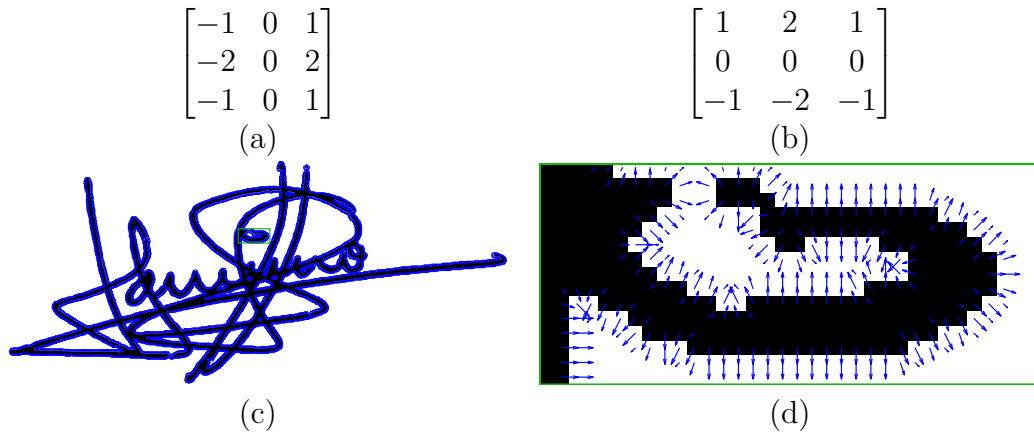


Figure 2.5: The Sobel filter kernel used for the (a)  $x$  and (b)  $y$  derivative. (c) A signature image with the gradient vectors (blue) computed from the two derivatives. (d) A magnification of the green area marked in (c).

and columns with less than two pixels have been removed from the edge of the image.

- Baseline shift – the difference between the vertical centres of gravity in the left and right half of the image.
- Vertical centre of the signature – the vertical centre of gravity obtained from the horizontal projection of the image.
- Horizontal centre of the signature – the horizontal centre of gravity obtained from the vertical projection of the image.
- Maximum vertical and horizontal projection – the maximum value of the vertical and horizontal projection histogram, respectively.
- Global slant angle – the rotation angle at which the horizontal projection reaches its maximum. It is obtained by rotating the image by  $-30^\circ$  and then rotating it back in steps of  $2^\circ$  and computing the horizontal projection at each step.

These features together with the number of edge points, the number of cross points, the number of closed loops and the grid feature form the final feature vector.

Two approaches based on high-level features are presented by Oz [39] and Shirdhonkar and Kokare [48]. Oz uses moment invariants as proposed by Hu [22] and computes them at 12 different angles (i.e. in  $30^\circ$  steps from  $0^\circ$  to  $360^\circ$ ).

The two-dimensional moments of a geometrical pattern which is represented by a density distribution function  $\rho(x, y)$  is defined as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \rho(x, y) \, dx dy, \quad (2.1)$$

where  $(p + q)$  is the order of the moment and  $p, q \in \mathbb{N}_0$  [22]. The central moments are then defined as

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q \rho(x, y) \, d(x - \bar{x}) d(y - \bar{y}), \quad (2.2)$$

with  $\bar{x} = m_{10}/m_{00}$  and  $\bar{y} = m_{01}/m_{00}$ . From these moments Hu [22] derives the 7 moment invariants shown in Table 2.1. This results in a feature vector of 84 values (i.e. 7 invariants at 12 different angles) for each signature that is independent of size, orientation and translation [39]. While yielding good recognition results (i.e. 100% success rate) for known signatures the moment invariants perform poorly for signatures which were not in the training set. However, Oz overcomes this problem by applying signature verification after the recognition step to sort out false results.

Table 2.1: Moment invariants as used by Oz [39].

---

$I_1 =$	$\mu_{20} + \mu_{02}$
$I_2 =$	$(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$
$I_3 =$	$(\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$
$I_4 =$	$(\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$
$I_5 =$	$(\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12}) \cdot [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03}) \cdot [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$
$I_6 =$	$(\mu_{20} - \mu_{02}) \cdot [(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$
$I_7 =$	$(3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12}) \cdot [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03}) \cdot [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$

---

Shirdhonkar and Kokare [48] use wavelets for signature retrieval where it is not possible to rely on a verification step to eliminate false matches. They use Dual Tree Complex Wavelet Transforms (DT-CWTs) and Dual Tree Rotated Complex Wavelet Filters (DT-RCWFs), which are obtained by rotating DT-CWT filters by  $45^\circ$  [49], to decompose each signature into energy and standard deviation components. Energy  $E_k$  and standard deviation  $\sigma_k$  at sub-band level  $k$  are computed as

$$E_k = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N |W_k(i, j)|, \quad (2.3)$$

$$\sigma_k = \sqrt{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (W_k(i, j) - \mu_k)^2}, \quad (2.4)$$

where  $W_k(i, j)$  is the wavelet-decomposed sub-band at level  $k$ ,  $M \times N$  is the size of this sub-band and  $\mu_k$  is the mean at sub-band level  $k$  with  $k = 1, \dots, 6$ . The final feature vector for each signature image is obtained by unifying the energy and standard deviation values of all sub-bands.

### 2.3.2 Verification and Identification

Han and Sethi [21] who use string representations as features for their signatures perform signature identification by storing all strings in one hash table per axis (horizontal and vertical). They use a perfect hash function to map each sequence of 3 consecutive characters of a string to a position in the hash map where the ID of the corresponding signature is stored in a linked list. To identify the author of a query signature, first the hash map addresses are computed for its strings. The IDs stored at each address vote for their corresponding signature and the top 3 signatures from both hash maps become candidate signatures. Finally, the Longest Common Subsequence (LCS) is computed for each candidate signature and the candidate with the longest LCS is used to identify the author of the query signature. To account for unknown query signature authors, the identification is only valid if the length of the LCS is above a threshold. The drawback of this approach is that it requires the administration of two hash maps and several linked lists.

Pavlidis et al. [40] propose a method with less complex data structures which uses credit points to find the author of a query signature based on its string representation. For two strings  $D = d_1 d_2 \dots d_n$  and  $E = e_1 e_2 \dots e_m$ ,  $d_k = e_j$  is considered a *full match* which yields two credit points and  $d_k \pm 1 = e_j$  is considered a *half match* that is worth one credit point. The strings are aligned by the characters representing the first angle on the left of the starting point of the feature extraction stage. Based on the credit points  $H$  they compute a similarity measure  $Q$  which is infinite for a perfect match and zero if  $H = 0$  as

$$Q = \frac{H}{2 \cdot \max(|D|, |E|) - H}. \quad (2.5)$$

The candidate string with the largest value of  $Q$  is used to identify the query signature. Again a threshold is used to verify that the identification is valid.

The following sections give an overview of some of the widely used machine learning methods for deciding whether a signature is genuine or to identify its author. This includes NNs, Hidden Markov Models (HMMs) and SVMs.

## Neural Networks

An artificial NN is a statistical model that tries to emulate a brain [55]. It consists of an input and an output layer and one or more hidden layers which contain several neurons. The neurons between layers have weighted connections and process the input of the preceding layer using a transfer function. The result is passed on to the next layer until it reaches the output layer. Before the NN can be used to verify or identify a signature it has to be trained, i.e. the weights of the connections have to be adjusted such that the NN generates the desired output for a set of training data.

Oz [39] uses two similar NNs for verification and identification. Both NNs have 84 input variables (the moment invariants) and one hidden layer. For verification the hidden layer has 40 neurons and the output layer has 2 variables for indicating a genuine or fake signature. The NN used for identification has 60 neurons in the hidden layer and 40 in the output layer (one value per signer). Karki et al. [26] also use NNs for verification and identification. They use 20 neurons for the input layer and 10 in the hidden layer with a log-sigmoid transfer function. The NN has 20 output values each corresponding to a signer. Oz and Karki et al. both use the backpropagation algorithm to modify the weights such that the difference between the desired output and the output of the NN is minimized.

## Hidden Markov Models

An HMM is a statistical model that is used for pattern recognition by computing the likelihood that a given output (i.e. a feature vector) is produced by a certain HMM. Each HMM consists of several states and transitions between the states with certain probabilities. After each transition an output value is produced by the current state [43]. Given a feature vector it is thus possible to compute the likelihood for each HMM that this model produces the output sequence represented by the feature vector. In signature identification there is usually one HMM for each signature [8]. A query signature is therefore identified as a signature of the author of the candidate signature which belongs to the HMM with the highest probability of producing the given feature vector as its output. A simple example of an HMM with four states is given in Figure 2.6. The transitions between the states are illustrated with black arrows where the saturation indicates the probability for a certain transition. Each state produces one of two output values with a certain probability which is illustrated using green arrows with different saturations.

## Support Vector Machines

SVMs were originally designed to find the hyperplane which separates two classes with maximum margin [37]. This hyperplane can be found by considering only a

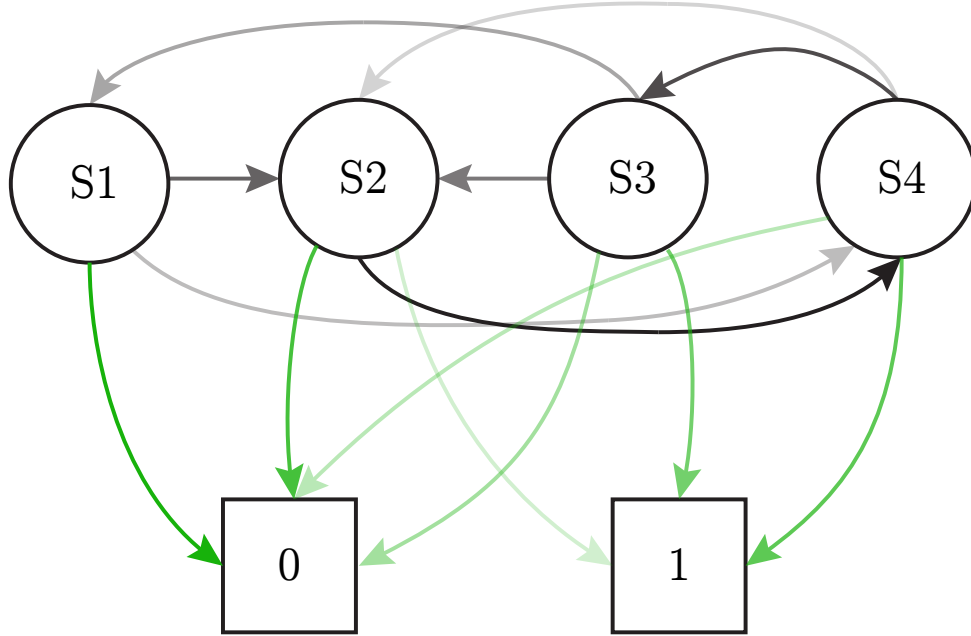


Figure 2.6: Example of a simple HMM for binary feature vectors with colour-coded state transition (black) and output (green) probabilities (higher saturation corresponds to a higher probability).

small subset of training data, i.e. the so-called support vectors [11]. It can be used to decide to which of the two classes the feature vector of a query signature belongs. In order to separate training data that cannot be linearly separated it is possible to use an SVM with a non-linear kernel function (e.g. an RBF). For data sets that cannot be separated even with a non-linear kernel function so-called *slack variables* are used to represent the magnitude of the classification error. Since SVMs are only able to make two-class decisions, it is necessary to use multiple SVMs for multi-class problems. Two typical strategies are one-against-all, where one SVM is used per class to separate this class from all other classes, and one-against-one which uses one SVM for each pair of classes [37]. Justino et al. [25] successfully use SVMs in signature verification where one SVM can be used per individual to decide whether a signature is genuine or forged.

### 2.3.3 Retrieval

In signature retrieval there is no prior knowledge of classes as it is in the case of signature verification and identification. Thus the desired output of a signature retrieval system is not a class label or a flag which indicates whether a signature

is genuine or not, but a list of signatures ranked by their similarity. Although this means that signature verification and identification techniques cannot be used for retrieval, it is possible to modify some methods for this purpose. For example, a signature identification method that is based on a nearest neighbour classifier (i.e. it assigns a query signature the class label with the smallest distance) can also be used to produce a ranking of similar signatures based on their distances [21].

The ranking distances for image retrieval are computed from the feature vectors using a distance function. Such a distance measure or metric  $D$  for vectors  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  must have the following properties [15]:

- $D(\mathbf{a}, \mathbf{b}) \geq 0$  (non-negativity)
- $D(\mathbf{a}, \mathbf{b}) = 0$  if and only if  $\mathbf{a} = \mathbf{b}$  (reflexivity)
- $D(\mathbf{a}, \mathbf{b}) = D(\mathbf{b}, \mathbf{a})$  (symmetry)
- $D(\mathbf{a}, \mathbf{b}) + D(\mathbf{b}, \mathbf{c}) \geq D(\mathbf{a}, \mathbf{c})$  (triangle inequality)

The rest of this section presents state-of-the-art distance measures and discusses their area of application.

Srihari et al. [50] who use binary feature vectors compute the distances of a query signature to all other signatures by means of a correlation similarity measure. For two feature vectors  $X$  and  $Y$  the similarity measure  $S$  is computed as

$$S(X, Y) = \frac{1}{2} + \frac{S_{11}S_{00} - S_{10}S_{01}}{2\sqrt{(S_{10} + S_{11})(S_{01} + S_{00})(S_{11} + S_{01})(S_{00} + S_{10})}}, \quad (2.6)$$

with  $S_{ij}$  denoting the number of occurrences where pattern  $i$  appears at the same position in vector  $X$  as pattern  $j$  appears in vector  $Y$ . The retrieval results are then obtained by ordering the signature images according to their similarity measure, starting with the smallest distance. The drawback of this similarity measure is that it can only be used for binary feature vectors.

A distance measure for general feature vectors is used by Shirdhonkar and Kokare [48] who compute the distance between the feature vector  $x$  of a query signature and the feature vector  $y$  of a candidate signature using Canberra distance, which is defined as

$$\text{Canb}(x, y) = \sum_{i=1}^d \frac{|x_i - y_i|}{|x_i| + |y_i|}, \quad (2.7)$$

with  $d$  denoting the length of a feature vector. They improve the retrieval results using a user-guided relevance feedback algorithm that moves a query closer to relevant signature samples and further away from irrelevant ones.

Another approach proposed by Zhu et al. [61] uses four dissimilarity measures that are computed from the transformation of a query signature to a candidate

signature. They combine the four dissimilarity measures to a final distance which is then used to rank the signature images accordingly. The retrieval results are further improved by using multiple signature images of one person as queries and combining their distances. A more detailed description of their approach is given in Chapter 3.

A distance measure that is used to compare histograms is the  $\chi^2$  test statistic. It gives less importance to differences between large bins than to differences between small bins [41]. For two histograms  $P, Q$  it is defined as

$$\chi^2(P, Q) = \frac{1}{2} \sum_i \frac{(P_i - Q_i)^2}{P_i + Q_i}, \quad (2.8)$$

where  $i$  specifies the bin in each histogram. However, this histogram distance measure only takes bin-to-bin differences into account.

The Earth Mover's Distance (EMD) on the other hand also considers cross-bin relationships as it is based on solving the transportation problem [56]. It is defined as the minimal cost of transforming one histogram into another and is computed as the work that is necessary for the transformation divided by the optimal flow that solves the transportation problem [45]. For two histograms  $P, Q$  it is given as

$$\text{EMD}(P, Q) = \frac{\sum_{i,j} d_{ij} f_{ij}}{\sum_{i,j} f_{ij}}, \quad (2.9)$$

where  $f_{ij}$  is the flow between  $P_i$  and  $Q_j$  that minimizes the overall cost,  $d_{ij}$  is the distance between  $P_i$  and  $Q_j$ , and the following constraints hold:

$$f_{ij} \geq 0, \sum_j f_{ij} \leq P_i, \sum_i f_{ij} \leq Q_j, \sum_{i,j} f_{ij} = \min \left( \sum_i P_i, \sum_j Q_j \right). \quad (2.10)$$

While the EMD can be both robust and discriminative [41], it is computationally more complex than the  $\chi^2$  distance.

## Summary

This chapter gave an overview of the steps involved in a document image retrieval system which is based on handwritten signatures. It presented state-of-the-art methods for signature localization, preprocessing and signature matching. Each part also included related topics as handwritten text localization and signature identification and verification to provide a broad perspective on the topic. It was shown that signature localization remains an open research area and that the majority of papers on signature matching does not deal with localization [16, 61].



The last section of this chapter presented machine learning methods that are successfully used in signature verification and identification. However, it was also explained that these techniques cannot be directly used for signature retrieval since it requires a ranking based on distance or similarity values. Hence, the methodology presented in the following chapter of this thesis is also based on a distance measure, namely the approach proposed by Zhu et al. [61].

## Methodology

The signature retrieval system proposed in this thesis is mainly based on the methods presented by Zhu et al. [61] but also introduces modifications that result in reduced computational time and increased matching performance. The main difference is the use of a shape-context-based pre-filtering step that reduces the computational time by a factor of 16.

In the first step of the retrieval system the image is preprocessed as proposed by Lin and Chang [33]. This step normalizes the image and extracts an abstract representation of the signature. The point set which represents the signature in the remainder of the algorithm is created by randomly sampling points on the abstraction of the signature image. In the next step the shape context descriptor [2] is computed for each point set and is subsequently used to compute the shape context distance to the remaining signature images in the test set. This distance is used in the following pre-filtering step to decide whether the image is processed further or not. In the former case the TPS transformations which best map the point set to the point sets of the other images are computed. Each TPS transformation is then used to compute four distance measures which accumulate to the overall distance of the signature to another in the test set through a weighted sum. The weights that are used to combine the distance measures are obtained using LDA. The retrieval is finally performed by ranking the shape context distances of the filtered images and the combined distance measures of the remaining images. The workflow of the signature retrieval system is illustrated in Figure 3.1. The first row shows the points sampled on the abstract representation of the signature and their distribution into different bins of the shape context descriptor for an example point. The shape context descriptor is used to filter the set of signatures such that the steps in the second row (i.e. the computation of the TPS transformation and the four dissimilarity measures) do not have to be computed for all images in the

set. The results of the first and second row are combined to rank the signature images in the set as depicted on the right.

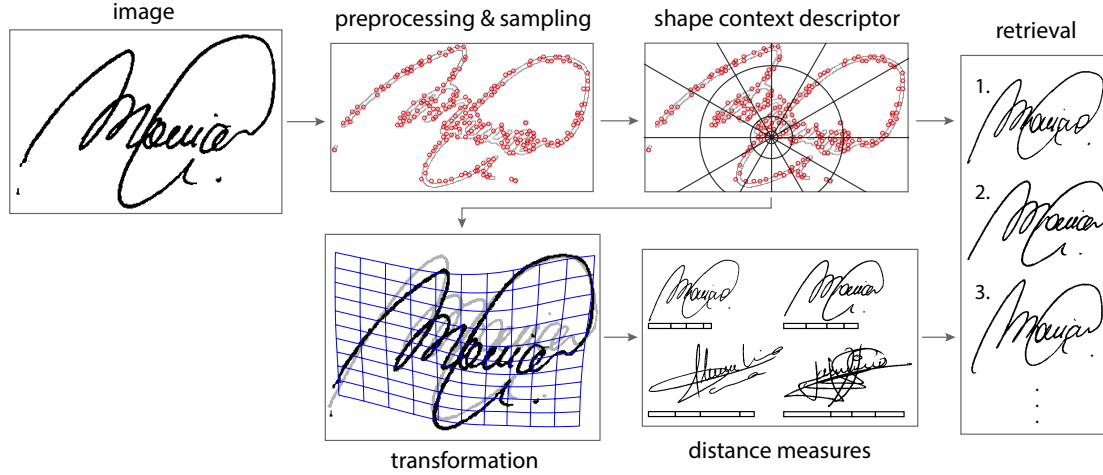


Figure 3.1: The workflow of the signature retrieval system. The steps in the second row are only performed for the signatures which remain after filtering. The results of the steps in both rows are combined in the retrieval step.

This chapter gives a detailed description of the techniques that are used in the signature retrieval system that is developed for this thesis. Section 3.1 gives an introduction to key technologies that are used in the remainder of this chapter. The preprocessing of the signature images and the extraction of the point set that is used for further processing is described in Section 3.2. The computation of the TPS transformation that best maps one point set to another is described in Section 3.3 and Section 3.4 explains the computation of the four distance measures. Finally, Section 3.5 describes the pre-filtering process.

## 3.1 Theoretical Background

This section presents techniques and technologies that are used in the signature retrieval system proposed in this thesis and explains their theoretical backgrounds. It is intended as an introduction for readers who are not familiar with these topics.

### 3.1.1 Thin-Plate Splines

The signature retrieval algorithm in this thesis uses TPSs to model the transformation from one point set to another. The TPS transformation is also used to compute the four distance measures which accumulate to the final distance. This

section explains how they are used to support the matching algorithm by assessing the quality of correspondence assignments and also gives a brief overview of their underlying theory.

TPSs are a tool for interpolating surface splines over scattered data [5]. The algebra is based on the physical properties of an infinite thin metal plate that is fixed at certain points (i.e. the control points). This plate fits the control points such that the energy necessary to bend it is minimized. When used as a transformation for two-dimensional point matching, the amount of energy that is necessary to deform the TPS such that one point cloud matches the other can be used as an indicator for the quality of the match. This energy – the so-called *integral bending norm* – is measured using only the non-affine part of the transformation, which is possible since the TPS is able to model affine and non-rigid transformations such that they can be separated. The remainder of this section describes the algebra of TPSs and the computation of the integral bending norm as presented by Bookstein [5].

The TPS transformation is described using the function

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^n w_i U(\|P_i - (x, y)\|), \quad (3.1)$$

where  $a_1, a_x, a_y$  are the affine coefficients,  $w_i$  are the non-affine coefficients,  $U(r) = r^2 \log r^2$  and  $P_i$  with  $i = 1, \dots, n$  are the control points the TPS passes through. The computation of the coefficients is done using the matrix

$$L = \left[ \begin{array}{c|c} \Phi & P^h \\ \hline P^{h\top} & \mathbf{0} \end{array} \right], \quad (3.2)$$

with the  $n \times n$  TPS kernel matrix

$$\Phi = \begin{bmatrix} 0 & U(r_{12}) & \cdots & U(r_{1n}) \\ U(r_{21}) & 0 & \cdots & U(r_{2n}) \\ \cdots & \cdots & \cdots & \cdots \\ U(r_{n1}) & U(r_{n2}) & \cdots & 0 \end{bmatrix}, \quad (3.3)$$

for  $r_{ij} = \|P_i - P_j\|$ , the set of control points in homogeneous coordinates

$$P^h = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix}, \quad (3.4)$$

and the  $3 \times 3$  matrix of zeros  $\mathbf{0}$ . The TPS coefficients are then computed from a vector  $V_0 = (V, 0, 0, 0)$  with  $V = (v_1, \dots, v_n)$  as

$$L^{-1}V_0 = (w_1, \dots, w_n, a_1, a_x, a_y). \quad (3.5)$$

By setting  $V = (x'_1, \dots, x'_n)$  and  $V = (y'_1, \dots, y'_n)$  with  $(x'_i, y'_i)$  being the coordinates of a second point set that correspond to  $(x_i, y_i)$  this yields the coefficients of the functions  $f_x, f_y$  describing the  $x$  and  $y$ -coordinate changes respectively.

As mentioned before, function  $f$  minimizes the integral bending norm which is given as

$$I_f = \iint_{\mathbb{R}^2} \left[ \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 \right] dx dy. \quad (3.6)$$

The value of  $I_f$  is proportional to

$$W \Phi W^\top = V \underbrace{(L_n^{-1} \Phi L_n^{-1})}_B V^\top, \quad (3.7)$$

where  $W = (w_1, \dots, w_n)$  and  $L_n^{-1}$  is the upper left  $n \times n$  block of the inverse matrix of  $L$ . Matrix  $B$  is also called the bending energy matrix since its eigenvectors indicate the amount of bending at each control point.

An example with two small sets of 5 points each is given in Figure 3.2. The point sets  $P$  in Figure 3.2 (a) and  $Q$  in Figure 3.2 (b) are

$$P = \begin{bmatrix} 3.6929 & 4.8189 & 6.7756 & 5.6969 & 6.5827 \\ 10.3819 & 11.2047 & 12.0866 & 10.0748 & 8.8386 \end{bmatrix}$$

and

$$Q = \begin{bmatrix} 3.9724 & 5.4016 & 6.5394 & 5.7756 & 6.6969 \\ 6.5354 & 6.4528 & 7.2362 & 5.1142 & 4.1181 \end{bmatrix},$$

where the  $i$ th point in set  $P$  corresponds to the  $i$ th point in set  $Q$  for  $i = 1, \dots, 5$ . The bending energy matrix as defined in Equation 3.7 has two eigenvectors that correspond to non-zero eigenvalues for this example. These eigenvectors are

$$(-0.2152, 0.6553, -0.1346, -0.6320, 0.3265)$$

and

$$(-0.4941, 0.4700, -0.3370, 0.6026, -0.2415)$$

and can be interpreted as the amount of bending that is necessary at each control point. It can be observed that the sum of absolute energy necessary is highest at points 2 ( $0.6553 + 0.4700$ ) and 4 ( $0.6320 + 0.6026$ ) which matches the visual impression of Figure 3.2 (b).

### 3.1.2 Shape Context Descriptor

The shape context descriptor is a rich descriptor of the shape of an object which was introduced by Belongie et al. [2]. It is a central component of the retrieval system

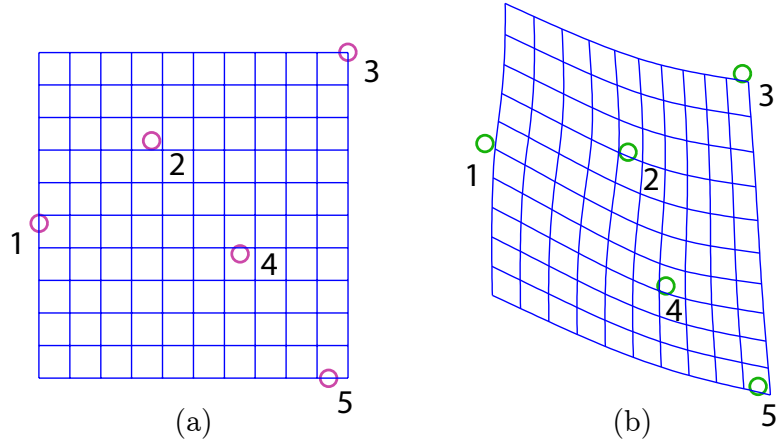


Figure 3.2: (a) Point set  $P$  with an undistorted TPS grid. (b) Point set  $Q$  together with the TPS transformation that maps  $P$  to  $Q$ . The numbers indicate the correspondences between the points in the different sets. Figure inspired by [5].

proposed in this thesis as it is used in the pre-filtering step as well as to compute one of the four distance measures. It describes a shape by connecting each sample point on the shape with all other sample points. Since different representations of shapes may vary from one instance to another, the entire graph would be much too detailed [2]. This is why Belongie et al. use the lengths and orientations of the connecting lines to create a log-polar histogram for each sample point. This histogram typically has 12 bins for the orientations and 5 bins for the logarithm of line lengths. The logarithm is applied to give more emphasis to changes in the distance of close points than to points which are far apart. This representation effectively describes the structural relation of one point to the other points in the set and can therefore be used to evaluate the quality of a match. An example is given in Figure 3.3. The first row shows two point sets and the bin distribution in the log-polar histogram. The second row depicts the shape context histograms of the three points which are marked in the point sets. It shows that the histograms of two points which are in a similar position on a similar shape (i.e. circle and diamond) are also similar, while the histogram of a point in a different position looks different.

Since the shape context histogram of each point captures the relative position of other points with respect to this point, changes in the orientation of the image result in changes in the histogram. The shape context descriptor is therefore not orientation invariant. The histogram bins which store the logarithmic lengths of connecting lines, however, are normalized by the line lengths, thus making the descriptor invariant to scale. This behaviour is illustrated in Figure 3.4, where (c)

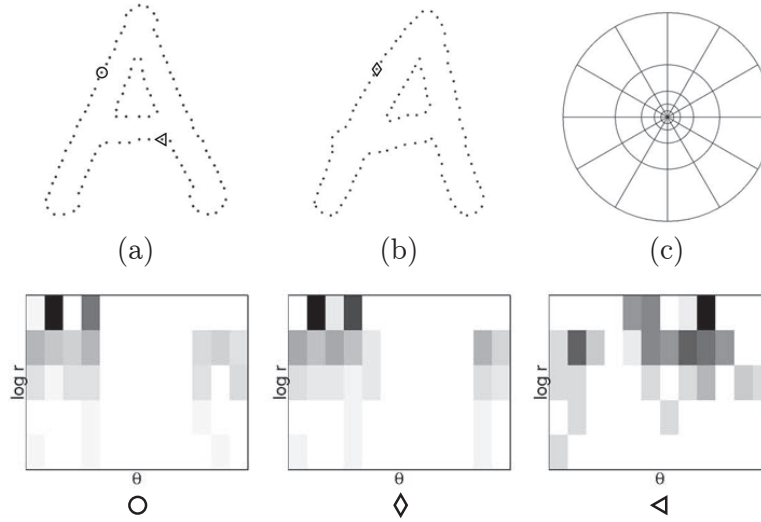


Figure 3.3: (a) and (b) show two point sets sampled from different shapes of a character. (c) shows the bins of the log-polar histogram. The second row shows the shape context histograms of the points which are marked in the sets in the first row. The correspondences are indicated by the symbols (i.e. circle, diamond and triangle). Image taken from [2].

shows the image from (a) rotated by  $90^\circ$ . Figure 3.4 (d) and (f) show the shape context histograms which correspond to the point marked in (a) and (c) respectively. It can be observed that the bins are shifted horizontally due to the rotation, thus yielding a high shape context distance value of 70.25 for the two images. The shape context histogram in Figure 3.4 (e) which corresponds to the point marked in the scaled image in (b) on the other hand is exactly the same as the histogram for the same point marked in (a). The shape context distance between these images is therefore 0.

### 3.1.3 Linear Discriminant Analysis

Before the signature images can be ranked in the last step of the retrieval system the four distance measures have to be combined to a single distance for each image. This is accomplished by means of LDA which provides the respective weights for the distance measures. The remainder of this section discusses the basic principles of LDA and how it can be used to obtain the aforementioned weights.

In contrast to Principal Component Analysis (PCA) which tries to find directions for an efficient representation for data by means of dimensionality reduction, the goal of LDA [19] is to find directions that efficiently separate different classes

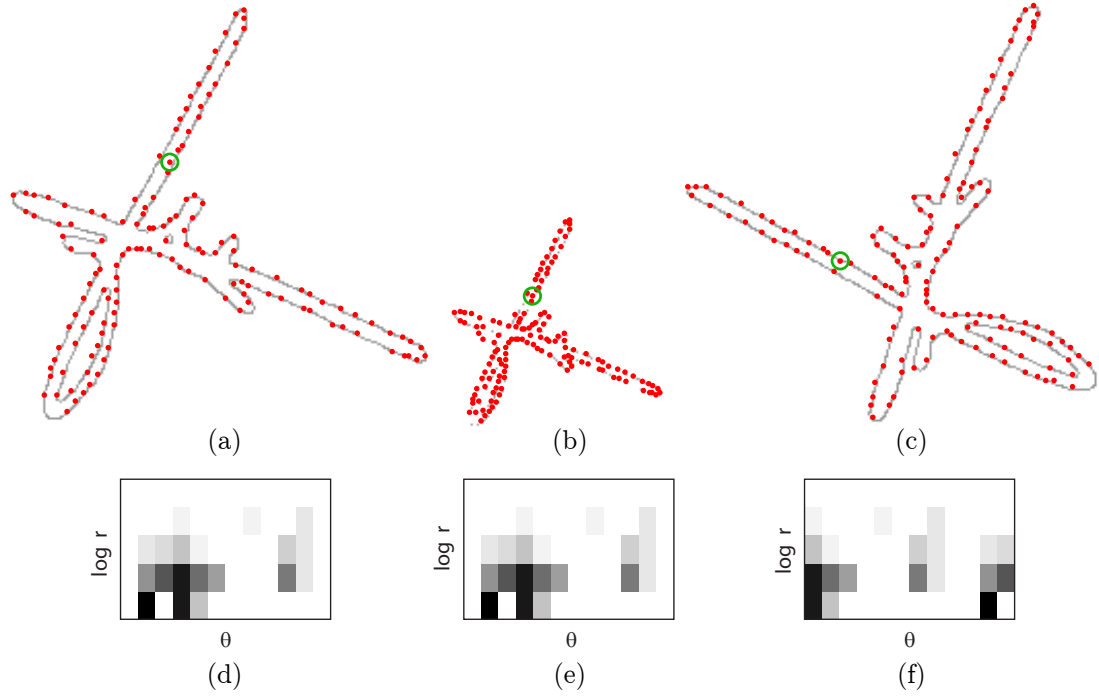


Figure 3.4: (a) Points sampled on a signature image. (b) The same image scaled by factor 0.5. (c) The image from (a) rotated by 90°. The second row shows the shape context histograms of the points which are marked by green circles in the first row.

of data [15]. This is done by projecting the data onto a line  $w$  and looking for the position and orientation of this line at which the separation of projected data is best. The separation is defined by a threshold that splits the points on the line into different classes. The border which separates the original data is then given as the hyperplane which is perpendicular to  $w$  and intersects it at the threshold.

This method is used in machine learning to train a linear classifier on a training set. The position and orientation of  $w$  is obtained from the normal distribution which is estimated for each class in the training set. The separating hyperplane is then used in the classification step to assign class labels to test data. As mentioned before, LDA can also be used to combine several features into one by taking the coefficients of the hyperplane as weights. An example is given in Figure 3.5 where (a) shows a two-dimensional example of the result of LDA on a set of training data and (b) shows one class separated from the other classes in the test set using the coefficients of LDA to combine four features into one.



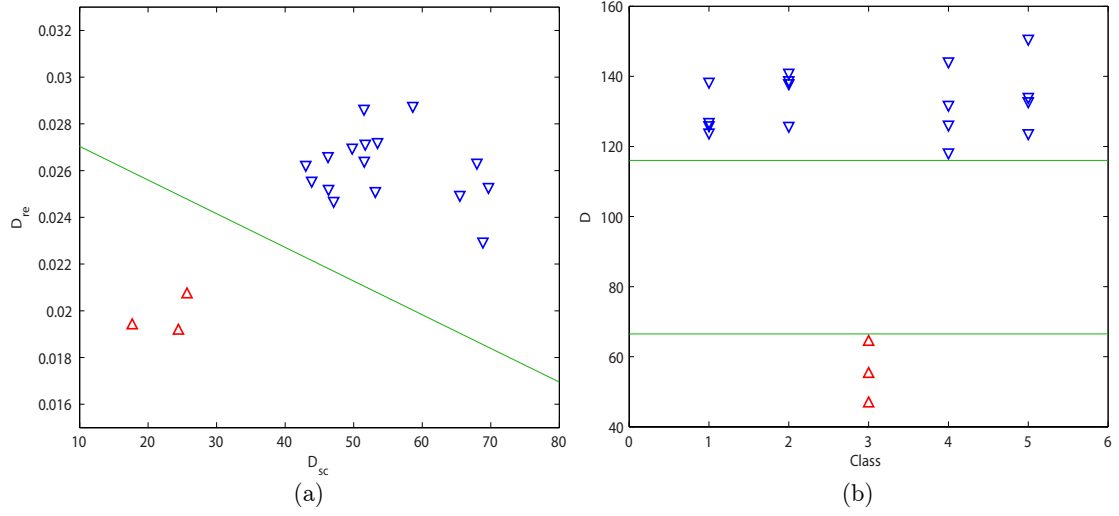


Figure 3.5: (a) Two-dimensional example of an LDA with one class (red) separated from another (blue) by a hyperplane (green line). (b) Coefficients of the separating hyperplane used to accumulate four features into one to separate one class (red) from four other classes (blue) in the training set. The margin between the two LDA classes is illustrated with two green lines.

## 3.2 Preprocessing and Point Sampling

Since the test set that is used for evaluating the methods in this thesis consists of noise-free, binary images of signatures it is not necessary to apply noise removal, printed text removal and binarization methods. Therefore the preprocessing in this thesis merely consists of a normalization step similar to the approach of Lin and Chang [33] and an abstraction step. In the first step the signature images are rotated such that the major axis of the Best-Fit Ellipse (BFE) is aligned with the horizontal axis. This step is necessary because the shape context descriptor is not invariant to changes in orientation (see Section 3.1.2). In contrast to Pavlidis et al. [40] the rotation is unambiguous in this case since there are no rotations  $\geq 90^\circ$  in the data set. It is therefore not necessary to store two representations of one signature image.

The BFE orientation normalization is compared to a shape-context-based and a gradient-based normalization in Section 4.2 to see which one performs best on the data set. The shape-context-based approach is chosen because it seems natural to use a descriptor that contains orientation information and is already used in other parts of the algorithm. It computes the shape context descriptor for the centre of gravity of all foreground pixels of an image using 18 bins for angles from  $-90$  to  $90$  degrees. Angles from  $90$  to  $270$  degrees are inverted such that they point in the

opposite direction and are counted as their  $[-90^\circ, 90^\circ)$  counterpart. The image is then normalized using the angle which corresponds to the largest bin.

The gradient-based orientation normalization is chosen because gradient orientations are also used to make the Scale-Invariant Feature Transform (SIFT) descriptor [34] invariant to rotation. This approach starts by convolving the image with a  $5 \times 5$  Gaussian filter with  $\sigma = 2$ . The  $x$  and  $y$  derivatives of the image are then approximated using two Sobel operators (see Figure 2.5). The gradient directions which are computed from these derivatives are again arranged in 18 bins for angles from  $-90$  to  $90$  degrees, similar to the shape-context-based approach, and the image is finally normalized with the angle that is represented by the largest bin. Figure 3.6 shows an example of the different results using these three orientation normalization techniques on skeleton images.

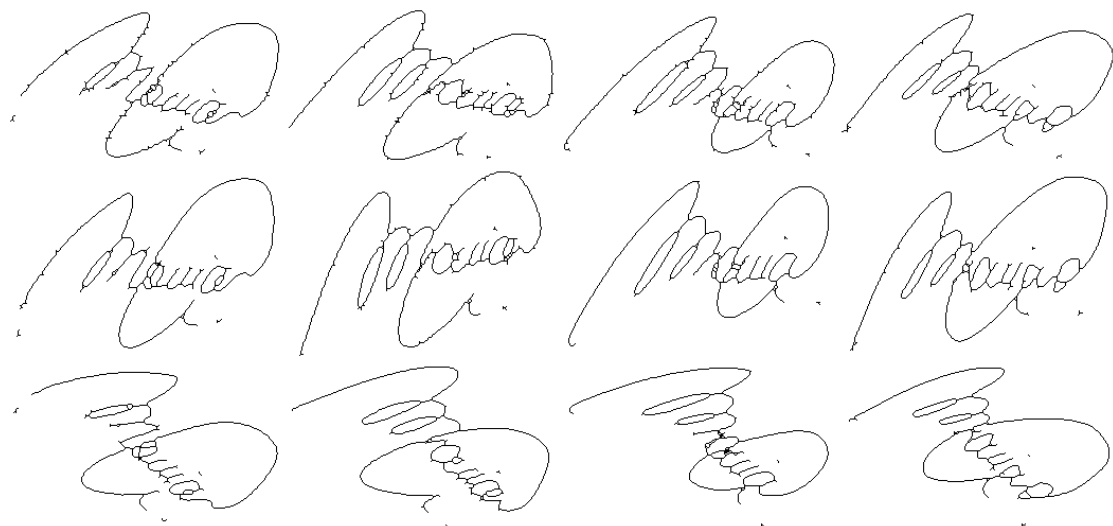


Figure 3.6: The results of using different orientation normalization techniques on skeleton images. The techniques are BFE (first row), shape context normalization (second row) and gradient normalization (third row).

After the orientation normalization step the image is trimmed to fit the size of the signature and resized to normalize the length of the diagonal (see Figure 3.7 (b)). The normalized diagonal length that is used for the evaluation in this thesis is set to 300 pixels.

Two different abstraction steps are used to evaluate their performance. The first technique uses a Canny edge detector to extract the contour of the signature (see Figure 3.7 (d)). This would not be necessary in a binary image where the contours are already given by neighbourhood relations among pixels. In this case, however, the image is not binary any more due to resampling in the scaling step. The second

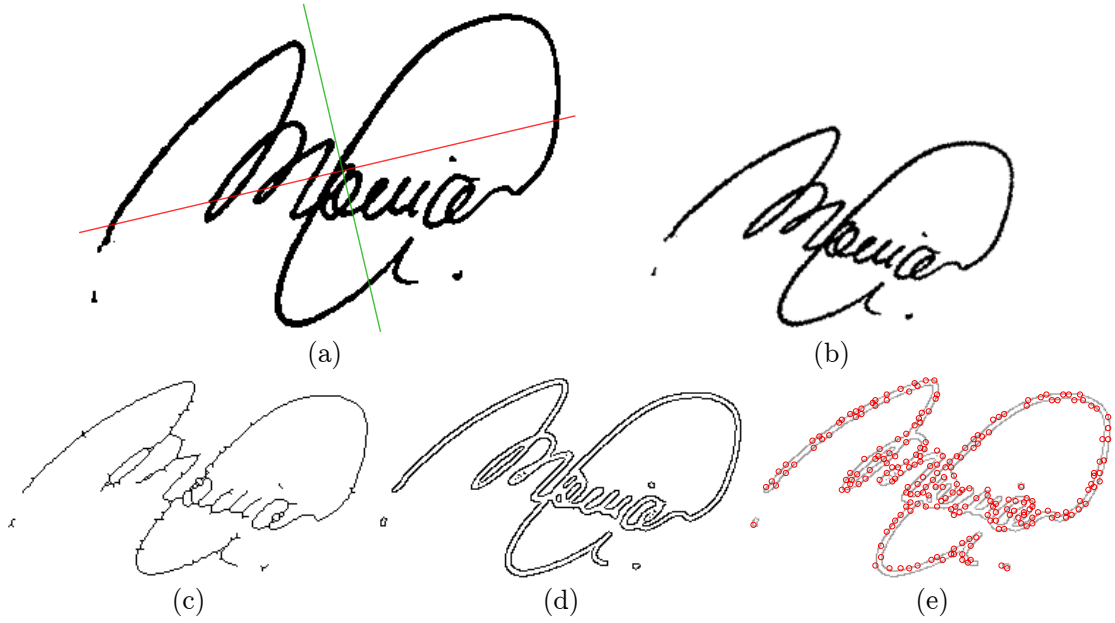


Figure 3.7: (a) Original signature image with major (red) and minor (green) axis of the BFE. (b) The same image after size and orientation normalization. (c) The skeleton of the normalized image. (d) The edges of the normalized image. (e) Points sampled on the edge image.

technique computes the skeleton of the signature using morphological thinning (see Figure 3.7 (c)) which requires a binary image to reduce the amount of spurs and artefacts in the resulting abstraction. Each image is therefore normalized to the interval  $[0, 1]$  and binarized by means of a simple thresholding with the constant  $c = 0.3$ .

The point set which represents the signature image in the remainder of the algorithm is created by randomly sampling points on the edge or skeleton image (see Figure 3.7 (e)) and normalizing them to the interval  $[0, 1]$ . In order to sample roughly the same amount of points for each image the sampling probability  $p$  for each pixel is computed using the black pixel density  $D_{bp}$ . It is computed as

$$p = \frac{q}{D_{bp}} = \frac{N \cdot q}{N_{bp}}, \quad (3.8)$$

where  $q$  is the sampling parameter which controls how many points are sampled,  $N_{bp}$  is the number of black pixels and  $N$  is the total number of pixels in the image. Each pixel that lies on the abstract representation of the signature and is selected for sampling, contributes to the final set of points which describes the signature.

For example, the point set for the signature in Figure 3.7 is computed using the sampling parameter  $q = 0.007$  which yields a sampling probability of  $p = 0.1$ . This means that every pixel in the image has a chance of 0.1 to be selected. Each selected pixel that overlaps with a foreground pixel of the signature is then added to the final point set (see Figure 3.8).

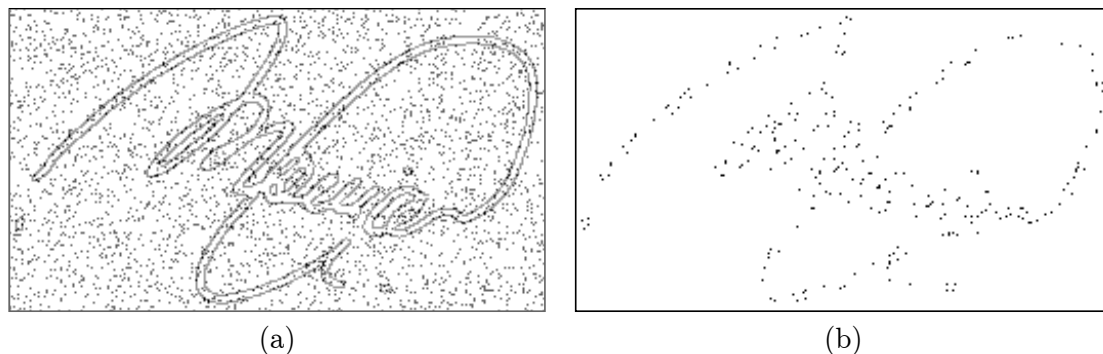


Figure 3.8: (a) All selected points in the image with  $p = 0.1$ . (b) The points which are selected and lie on the abstract representation of the signature, i.e. the final point set which represents the signature.

### 3.3 Thin-Plate Spline – Robust Point Matching Algorithm

The TPS transformation as described in Section 3.1.1 needs known correspondences between the points of two sets to be computed. To overcome this limitation Chui and Rangarajan [10] developed the so-called Thin-Plate Spline – Robust Point Matching (TPS–RPM) algorithm. It makes use of the fact that the computation of correspondences has an optimal solution once the transformation is known and vice versa [10] by computing them in an alternating manner until the solutions converge. Each step of this iterative process refines the correspondence assignment and the transformation.

The performance of the TPS–RPM algorithm is compared to the similar Iterative Closest Point (ICP) algorithm [3] by Chui and Rangarajan [10] using synthetic data to evaluate behavioural changes due to the degree of deformation, the amount of noise and the amount of outliers in the point sets. It is shown that the error rates of the TPS–RPM algorithm are not only lower than those of the ICP, but also increase much slower for test cases with increased deformation, noise and outliers.

The TPS–RPM algorithm uses an annealing parameter to influence the change of the TPS transformation such that it permits global changes in the beginning

and more local ones towards the end. This is done using a temperature variable  $T$  in analogy to physical annealing and allowing larger changes in the transformation for higher temperatures and smaller changes for lower ones. The temperature decreases at each annealing step according to the annealing rate  $r$  until the final temperature  $T_{\text{final}}$  is reached.

Instead of having a binary correspondence matrix Chui and Rangarajan choose a softassign approach where the correspondence values are continuous in the interval  $[0, 1]$ . This is due to the fact that binary correspondences are less meaningful in the early stages of the iteration because the transformation is still far away from the optimal solution [10].

The pseudo code of the algorithm is shown in Algorithm 1. The first step is to initialize the annealing parameters  $T$ ,  $T_0$ ,  $T_{\text{final}}$ , the annealing rate  $r$ , the TPS smoothness constraints  $\lambda_{1,2}^{\text{init}}$  and the TPS kernel matrix  $\Phi$ . For two point sets  $X$ ,  $V$  of size  $N$  and  $K$  respectively, the parameters are set to  $T = 0.5$ ,  $T_0 = \max(V_x^2)$ ,  $T_{\text{final}} = 0.001$ ,  $r = 0.93$ ,  $\lambda_1^{\text{init}} = 1$  and  $\lambda_2^{\text{init}} = \lambda_1^{\text{init}} \cdot 0.01$ , where  $V_x$  denotes the  $x$ -coordinates of point set  $V$ . The TPS kernel matrix used by Chui and Rangarajan differs slightly from that of Bookstein given in Equation 3.3 and is computed as  $\Phi_{ij} = ||V_i - V_j||^2 \log ||V_i - V_j||$  with  $V_i$ ,  $V_j$  being the coordinates of points  $i$  and  $j$  respectively.

---

**Algorithm 1** TPS-RPM Algorithm

---

- 1: initialize parameters and variables
  - 2: **while**  $T > T_{\text{final}}$  **do**
  - 3:   **for**  $i = 1 \rightarrow 5$  **do**
  - 4:     update correspondence matrix
  - 5:     update TPS parameters
  - 6:   **end for**
  - 7:   update annealing parameters
  - 8: **end while**
- 

The inner loop of Algorithm 1 consists of two parts. The first is to update the correspondence matrix  $m$  with the estimated transformation  $f(V)$  such that

$$m_{kn} = \frac{1}{\sqrt{T}} \exp \left( -\frac{(X_n - f(V_k))^T (X_n - f(V_k))}{T} \right), \quad (3.9)$$

for  $k = 1, \dots, K$  and  $n = 1, \dots, N$ . Since the transformation is unknown when  $m$  is computed for the first time it is initialized as  $f(V) = V$ . To account for outliers in the point sets,  $m$  is expanded by an outlier row and an outlier column:

$$m_{K+1,n} = m_{k,N+1} = \frac{1}{\sqrt{T_0}} \exp(-1). \quad (3.10)$$

The additional row and column are used to update the correspondences in  $m$  by first dividing each column and then each row by their sum respectively.

In the second part the TPS parameters are updated with the correspondences in  $m$ . For this purpose an additional point set  $Y$  is computed as

$$Y_k = \sum_{n=1}^N m_{kn} X_n, \quad (3.11)$$

which can be seen as the set of transformed points  $V$ . The TPS parameters are then computed using the homogeneous variants  $Y^h$  and  $V^h$  of these point sets where  $V_k^h = (1, V_{kx}, V_{ky})$  and  $Y_k^h = (1, Y_{kx}, Y_{ky})$ .  $V^h$  is then split up into the three matrices  $Q_1$ ,  $Q_2$  and  $R$  using QR decomposition [54] such that

$$V^h = [Q_1 \ Q_2] \begin{bmatrix} R \\ \mathbf{0} \end{bmatrix}, \quad (3.12)$$

where  $Q_1$  is a  $K \times 3$  matrix,  $Q_2$  is  $K \times (K - 3)$ ,  $R$  is  $3 \times 3$  and  $\mathbf{0}$  is a  $K \times 3$  matrix of zeros. Together with the smoothness constraints  $\lambda_1$ ,  $\lambda_2$  and the TPS kernel matrix  $\Phi$  these matrices yield the TPS parameters  $w$  and  $d$  as follows:

$$w = Q_2(Q_2^\top \Phi Q_2 + \lambda_1 I_{K-3})^{-1} Q_2^\top Y^h, \quad (3.13)$$

$$d = (R^\top R + \lambda_2 I_3)^{-1} (R^\top Q_1^\top (Y^h - \Phi w) - R^\top R) + I_3, \quad (3.14)$$

with  $I$  being a square identity matrix and its index indicating the size. The transformation of the sample points is then obtained using the function

$$f(V^h) = V^h d + \Phi w \quad (3.15)$$

and the whole process is repeated. Chui and Rangarajan [10] state that after five loop iterations the estimation is close to convergence which is why the inner loop stops after five runs. The annealing parameter  $T$  is subsequently updated by multiplying it with the annealing rate  $r$  and the smoothness constraints are updated using  $\lambda_{1,2} = \lambda_{1,2}^{\text{init}} \cdot T \cdot K$  respectively.

Figure 3.9 (a) shows the point sets that are matched (green dots and blue circles) and the transformed point set (red crosses) as result of the algorithm. Figure 3.9 (b) visualizes the TPS transformation as a blue grid on top of the two signatures that are matched.

### 3.4 Dissimilarity Measures

Once the transformation between the query signature and a candidate signature is known, it is used to compute four distance measures as proposed by Zhu et al. [61];

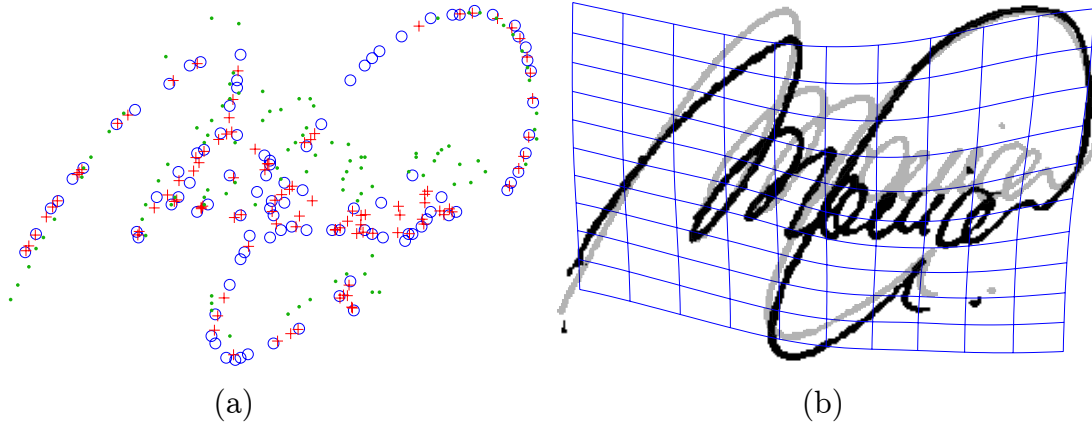


Figure 3.9: (a) The results of the TPS-RPM algorithm for finding a transformation from point set  $V$  (green dots) to  $X$  (blue circles). The transformed points  $f(V)$  are shown as red crosses. (b) The signatures from which the point sets are sampled together with the TPS transformation which is represented by a blue grid.

namely the bending energy  $D_{be}$ , the shape context distance  $D_{sc}$ , the anisotropic scaling  $D_{as}$  and the registration residual error distance  $D_{re}$ . They are accumulated into the final distance  $D$  using the weighted sum

$$D = w_{be}D_{be} + w_{sc}D_{sc} + w_{as}D_{as} + w_{re}D_{re}, \quad (3.16)$$

where the weights  $w$  are estimated via LDA on a random subset of signature images that are not in the test set. The weights are then obtained from the coefficients of the hyperplane (see Section 3.1.3) which separates the training set into two classes using a query signature which is also not in the test set.

Once the four distances for all signatures in the test set are obtained, the final distance  $D$  can also be computed without using training data by normalizing each distance measure with its standard deviation:

$$D = \frac{D_{be}}{\sigma_{be}} + \frac{D_{sc}}{\sigma_{sc}} + \frac{D_{as}}{\sigma_{as}} + \frac{D_{re}}{\sigma_{re}}. \quad (3.17)$$

### 3.4.1 Bending Energy

The integral bending norm of a TPS as described in Section 3.1.1 is a measure which relates to the amount of non-affine deformation in the transformation. In this thesis the variant of this norm is used that was proposed by Chui and Rangarajan [10] as

$$D_{be} = \lambda_1 \cdot \text{trace}(w \cdot (Y^h)^\top), \quad (3.18)$$

where  $\lambda_1$  is the TPS smoothness constraint of the non-affine part of the transformation,  $w$  is the TPS parameter describing the non-affine transformation (see Equations 3.13 and 3.15) and  $Y^h$  is the point set introduced in Equation 3.11.

### 3.4.2 Shape Context

The shape context descriptor (see Section 3.1.2) is used to compute the shape context distance  $D_{sc}$  between a set  $P$  from a query signature with  $k$  points and a set  $Q$  from a candidate signature with  $n$  points as stated in [2]:

$$D_{sc}(P, Q) = \frac{1}{k} \sum_{p \in P} \arg \min_{q \in Q} C(f(p), q) + \frac{1}{n} \sum_{q \in Q} \arg \min_{p \in P} C(f(p), q), \quad (3.19)$$

where  $f$  is the TPS transformation given in Equation 3.15 and  $C$  is the matching cost for two points, defined using the  $\chi^2$  test statistic:

$$C(p, q) = \frac{1}{2} \sum_{b=1}^B \frac{[h_p(b) - h_q(b)]^2}{h_p(b) + h_q(b)}, \quad (3.20)$$

where  $h_p$  and  $h_q$  are the shape context histograms of points  $p$  and  $q$ , and  $b$  specifies the bin with a total number of  $B$  bins.

### 3.4.3 Anisotropic Scaling

The anisotropic scaling is computed directly from the affine transformation matrix  $d$  (see Equations 3.14 and 3.15). It is defined in [61] as

$$D_{as} = \log \frac{\max(S_x, S_y)}{\min(S_x, S_y)}, \quad (3.21)$$

where  $S_x, S_y$  are obtained by singular value decomposition of  $d$ . Here  $S_x, S_y$  are the scaling factors of the affine part of the TPS transformation. Thus  $D_{as}$  is 0 if there is only isotropic scaling in  $d$  (i.e.  $S_x = S_y$ ).

### 3.4.4 Registration Residual Error

The last distance measure proposed by Zhu et al. [61] is the residual error of the estimated transformation. It describes the quality of the matching by computing the sum of Euclidean distances between corresponding points, normalized by the total number of matches. For a matching assignment  $M(i)$  it is defined as

$$D_{re}^H = \frac{\sum_{i=1}^{\min(k,n)} \|f(p_i) - q_{M(i)}\|}{\min(k, n)}, \quad (3.22)$$



where  $f$  is the TPS transformation given in Equation 3.15 and  $k, n$  are the sizes of point sets  $P$  and  $Q$  respectively. However, since this formula requires one-to-one correspondences and the TPS-RPM algorithm yields only soft matches (i.e. continuous values in the correspondence matrix instead of binary ones) it is first necessary to find correspondences. This is done using the Hungarian method [30] with the inverse matching quality from the correspondence matrix of the TPS-RPM algorithm as the matching cost.

Since the Hungarian method is computationally expensive, two other formulas for computing the registration residual error are suggested and evaluated in this thesis. The first one estimates correspondences by simply assigning the best match to each point of the smaller point set. Since this is likely to result in duplicate matches the following slightly modified formula is used to compute the registration residual error:

$$D_{re}^D = \frac{\sum_{i=1}^{\min(k,n)} \|f(p_i) - q_{M(i)}\|}{\min(k, n) - err(M)}, \quad (3.23)$$

where  $M(i)$  is the estimated matching assignment and  $err(M)$  is the number of duplicate matches in  $M$ .

The second alternative implementation does not estimate one-to-one correspondences at all, but simply uses the matching quality from the correspondence matrix of the TPS-RPM algorithm as a weight for the Euclidean distance between two points. It is defined as

$$D_{re}^W = \frac{\sum_{i=1}^k \sum_{j=1}^n m_{ij} \cdot \|f(p_i) - q_j\|}{\min(k, n)}, \quad (3.24)$$

where  $m$  is the correspondence matrix of the TPS-RPM algorithm (see Equation 3.9). The performance of the three variants of the registration residual error distance measure is evaluated in Section 4.1.

### 3.5 Pre-Filtering

Since the dissimilarity measures are computed from the transformation that best maps a query signature to a candidate signature, the time-consuming TPS-RPM algorithm has to be computed for the entire test set for each new query signature. Therefore it is suggested in this thesis to speed up the retrieval process by first reducing the search space. This is done by computing the shape context distance from a query signature to all other signature images in the test set similar to Equation 3.19 but without prior computation of the transformation (i.e.  $f(p) = p$ ). The results are then sorted and the expensive TPS-RPM algorithm and the dissimilarity measures are computed for only up to five percent of the highest

ranked signatures. The remaining signatures are ranked according to their shape context distance.

## Summary

This chapter presented the methodology that is used in the signature retrieval system which is proposed in this thesis. The system operates on point clouds which are sampled from preprocessed signature images using a probability function. These point clouds are matched using the TPS–RPM algorithm which also estimates the TPS transformation that maps one point cloud to the other. The TPS transformation is then used to compute four distance measures which are combined to a final distance value using weights that are estimated by means of LDA on a set of training data. The whole system is sped-up by introducing a pre-filtering step that reduces the amount of signature images for which the computationally expensive TPS–RPM algorithm and the distance measures are computed.

The following chapter uses the *GPDS960signature* dataset to evaluate the signature retrieval system and analyse the effects of using different orientation normalization and abstraction techniques (see Section 3.2), different sampling techniques and different methods for computing the registration residual error distance (see Section 3.4.4).

# CHAPTER 4

## Results

The signature retrieval system proposed in this thesis is evaluated in Matlab using the *GPDS960signature* database [4]. This database contains binary images of 24 genuine signatures from 960 individuals. It also contains 30 forged signatures for each individual, however, only genuine ones are used for the evaluation because this thesis does not deal with signature verification. Since the computation of the TPS-RPM algorithm and of the dissimilarity measures takes about 2.6 seconds for a single comparison without parallelization (i.e. about 16.6 hours for the evaluation of one query signature on the entire dataset of 960 signers and 24 signatures) an evaluation on the entire set is not feasible (see Table 4.1). The tests in this chapter are therefore conducted on the following subsets of this dataset:

*setA* – consists of 5 signatures each from 40 individuals, thus comprising a total of 200 signature images.

*setB* – consists of 8 signatures each from 120 individuals, thus comprising a total of 960 signature images.

The smaller set is used whenever the computation on the larger set would take too long (i.e. for the computations of the dissimilarity measures on their own unless they are compared to different approaches). Furthermore, the evaluation is parallelized on six cores since that is the maximum number of physical cores in the test system that can be used over a longer period of time (e.g. two weeks).

The performance of the document image retrieval system is evaluated using the same measures as in [61], namely Average Precision (AP) and R-Precision (RP). The precision of a retrieval system is computed as

$$\text{precision} = \frac{\text{number of relevant documents retrieved}}{\text{number of documents retrieved}}. \quad (4.1)$$

Table 4.1: Runtime comparison for a complete evaluation on different sets using parallelization for speed-up

Method	<i>setA</i>	<i>setB</i>	Full set
without pre-filtering	~18 hours	~17 days	~11 years
with pre-filtering	~1 hour	~1 day	~2 years

AP is the mean of the precisions at each rank that adds another relevant document, with a precision of zero for relevant documents that are not retrieved [6]. This means that the AP of a retrieval of a total of 3 relevant documents, where only 2 documents are found at positions 1 and 5, is given as  $AP = (1/1 + 2/5 + 0)/3 = 46.7\%$ . RP is the precision for retrieving R documents where R is the number of relevant documents for the given query. Thus the RP for the example given above is  $RP = 1/3 = 33.3\%$ , because only one relevant document is retrieved at rank 3. AP rewards higher rankings of relevant documents and penalizes that of irrelevant ones while RP ignores the exact ranking of the results and is more useful when a large number of relevant documents is present in the dataset [61].

All test runs in this chapter are conducted using each signature in the test set as query and removing it from the test set for this run. The average of the results for each query signature are then presented as the MAP and the MRP. Some of the results are also illustrated by plotting the average recall at each rank. The recall of a retrieval system is defined as

$$\text{recall} = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents}}. \quad (4.2)$$

As mentioned in Section 3.4, LDA is used to compute the weights for combining the four dissimilarity measures to a single distance (see Equation 3.16). Table 4.2 shows the actual weights that are utilized in the evaluation of the signature retrieval system.  $weight_1$  and  $weight_2$  are both obtained using a random training set that is one third of the size of *setB*, thus resulting in a 75% to 25% split between test set and training set. The training set contains at least two signature images from the same individual such that one is used as the reference image and the other is used to compute the distance and compare it to the distances to all the images from other individuals in the training set. The weights are then obtained from the hyperplane which separates these two classes as explained in Section 3.1.3. Since the training data for computing the weights are selected randomly, each computation yields different weights. Therefore, two weights are used in the evaluation such that each test run uses the weights that give the best overall results for this test.

The following sections evaluate each part of the retrieval system presented in Chapter 3 and also compare it to the performance of the dissimilarity measures on their own. First the three different registration residual error implementations

Table 4.2: Weights that are used to combine the dissimilarity measures

Name	$w_{be}$	$w_{sc}$	$w_{as}$	$w_{re}$
$weight_1$	50.31	0.1806	3.232	372.9
$weight_2$	52.99	0.1104	2.159	1,057

presented in Section 3.4.4 are evaluated to see which one performs best. Then the three orientation normalization techniques described in Section 3.2 are compared. The pre-filtering step and the dissimilarity measures are subsequently evaluated using edge detection and skeletonization as abstraction technique. The effects of varying sampling rates are then evaluated to find the best trade-off between retrieval performance and runtime. Finally, the performance of the signature retrieval system presented in this thesis is evaluated with different reduced set sizes and compared to the dissimilarity measures.

## 4.1 Registration Residual Error

The three different methods for computing the registration residual error distance as described in Section 3.4.4 are evaluated on *setA* for the sake of computation time. The test is performed using Canny edges as abstraction technique (see Section 4.3) and a sampling parameter of  $q = 0.007$  (see Equation 3.8) which yields about 200 points per signature image (see Section 4.4). Figure 4.1 illustrates the results in terms of MRP, MAP and total runtime and Table 4.3 shows the exact values. It can be observed that the two approaches proposed in this thesis (i.e.  $D_{re}^D$  and  $D_{re}^W$ ) are not only faster than the approach with the Hungarian method since they do not require the computation of optimal correspondences, but also achieve significantly higher retrieval rates. The method that is chosen for computing the registration residual error distance in the remainder of the evaluation is therefore  $D_{re}^W$ .

Table 4.3: Retrieval performances and runtimes of the different registration residual error implementations

Method	MRP	MAP	Runtime
$D_{re}^H$	5.1%	7.6%	25.4 h
$D_{re}^D$	33.6%	39.2%	16.4 h
$D_{re}^W$	48.3%	54.8%	15.6 h

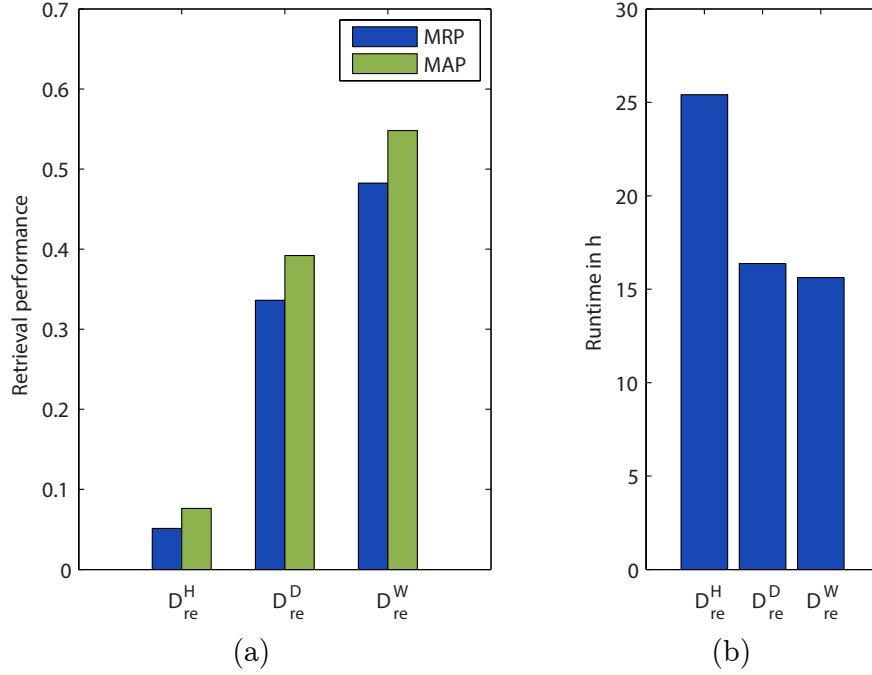


Figure 4.1: (a) The retrieval performances of the three implementations of the registration residual error distance in terms of MRP and MAP. (b) The runtimes of the evaluation of each implementation on the entire set of  $40 \times 5$  signature images in hours.

## 4.2 Orientation Normalization Techniques

The orientation normalization step is crucial in order for the pre-filtering to work properly (see Section 3.1.2). It is therefore important to evaluate which of the three techniques presented in Section 3.2 – i.e. BFE, shape context and gradient normalization – works best for the given data set. This is done by computing the shape context distance from each image in *setB* to all remaining images in the set and ranking the results accordingly. The results in terms of MRP, MAP and recall using skeletonization as abstraction technique are shown in Figure 4.2. The exact results are given in Table 4.4. It can be observed that the BFE achieves the highest retrieval rates and has the best recall with respect to the number of signature images retrieved, it is therefore the orientation normalization technique that is used for all other tests in this chapter.

Figure 4.2 (b) also shows that the gradient orientation normalization technique ranks more relevant signatures among the top ranks than the shape-context-based technique, but has to retrieve more images to find all relevant signatures. This

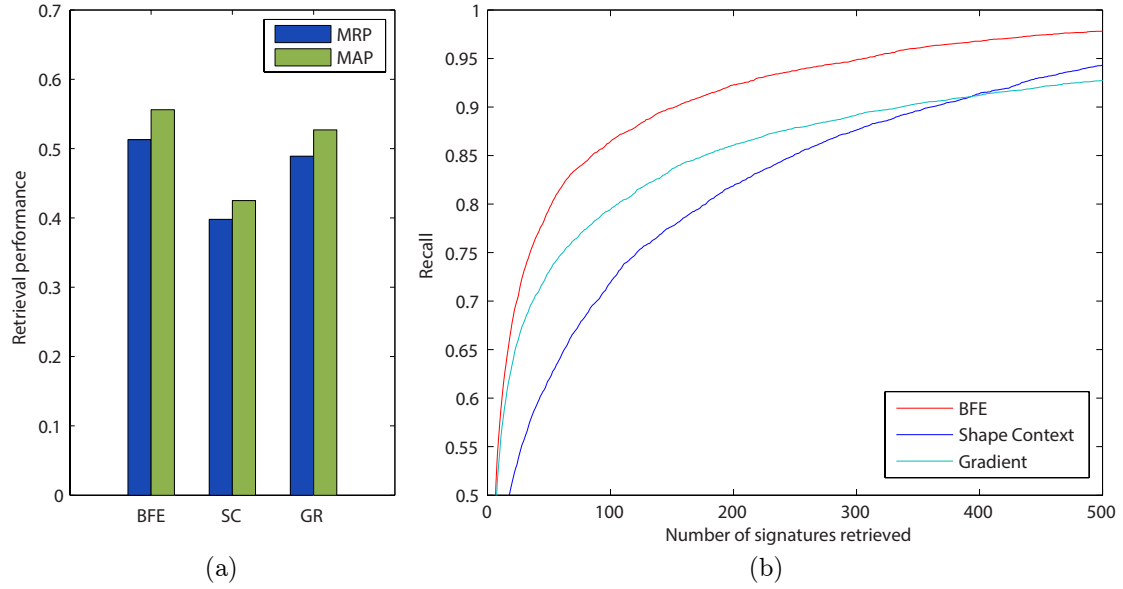


Figure 4.2: (a) The retrieval performances using the Best-Fit Ellipse (BFE), the Shape Context (SC) and the Gradient (GR) orientation normalization with skeleton images in terms of MRP and MAP. (b) The recall of the pre-filtering method using BFE (red), shape context (blue) and gradient normalization (cyan) on skeleton images with respect to the number of signature images retrieved.

Table 4.4: Retrieval performances of the different orientation normalization techniques using skeleton images

Method	MRP	MAP
BFE	51.3%	55.6%
Shape context	39.8%	42.5%
Gradient	48.9%	52.7%

means that it is better suited for finding as many relevant signatures as possible among the top ranks, but worse for finding all relevant signatures in a given set. The results for the test run on edge images are presented in Figure 4.3 and Table 4.5 and do not show great deviation from those for the tests using skeleton images.

### 4.3 Abstraction Techniques

As mentioned in Section 3.2 the retrieval system is evaluated using two different abstraction techniques (i.e. thinning or skeletonization and edge detection) to see

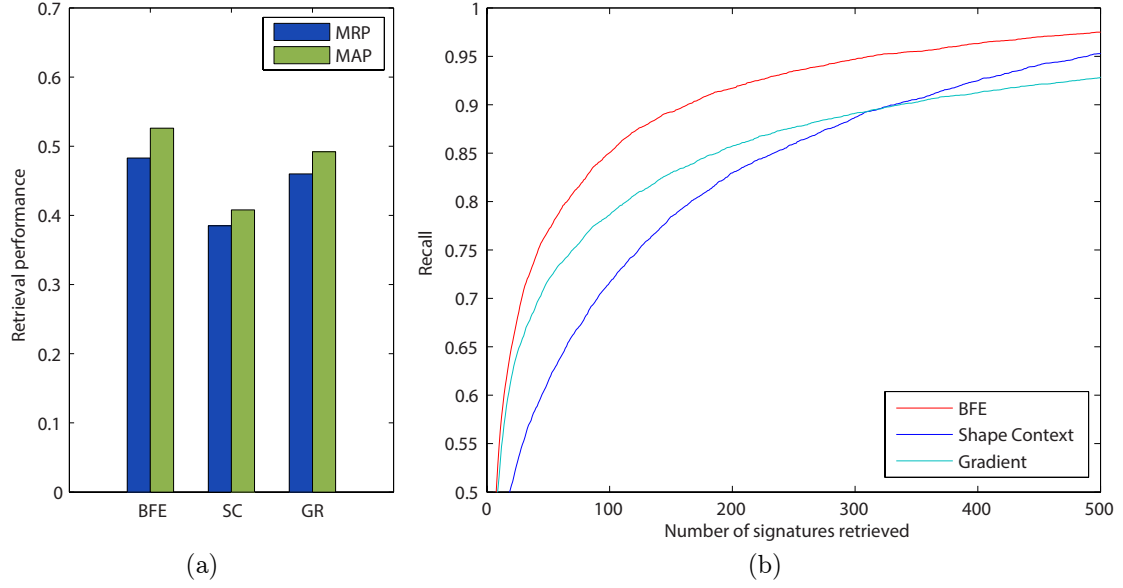


Figure 4.3: (a) The retrieval performances using the Best-Fit Ellipse (BFE), the Shape Context (SC) and the Gradient (GR) orientation normalization with edge images in terms of MRP and MAP. (b) The recall of the pre-filtering method using BFE (red), shape context (blue) and gradient normalization (cyan) on edge images with respect to the number of signature images retrieved.

Table 4.5: Retrieval performances of the different orientation normalization techniques using edge images

Method	MRP	MAP
BFE	48.3%	52.6%
Shape context	38.5%	40.8%
Gradient	46.0%	49.2%

how they perform. For this purpose the key parts of the retrieval system, namely the pre-filtering step using the shape context distance and the computation of the dissimilarity measures based on the TPS transformation, are both computed using skeleton and edge images. Instead of using the pre-filtering step to reduce the set for computing the dissimilarity measures, both parts are evaluated independently on *setB*. Figure 4.4 shows a comparison of the results for computing the shape context distance with both abstraction techniques in terms of (a) MRP, MAP and (b) recall with respect to the number of signatures retrieved. The comparison of the results for the dissimilarity measures is given in Figure 4.5. The exact results of the two test runs are given in Table 4.6 and 4.7 respectively. Both tests use a



sampling parameter of  $q = 0.007$  yielding about 200 points per signature image. The results for the dissimilarity measures are obtained using  $weight_1$ .

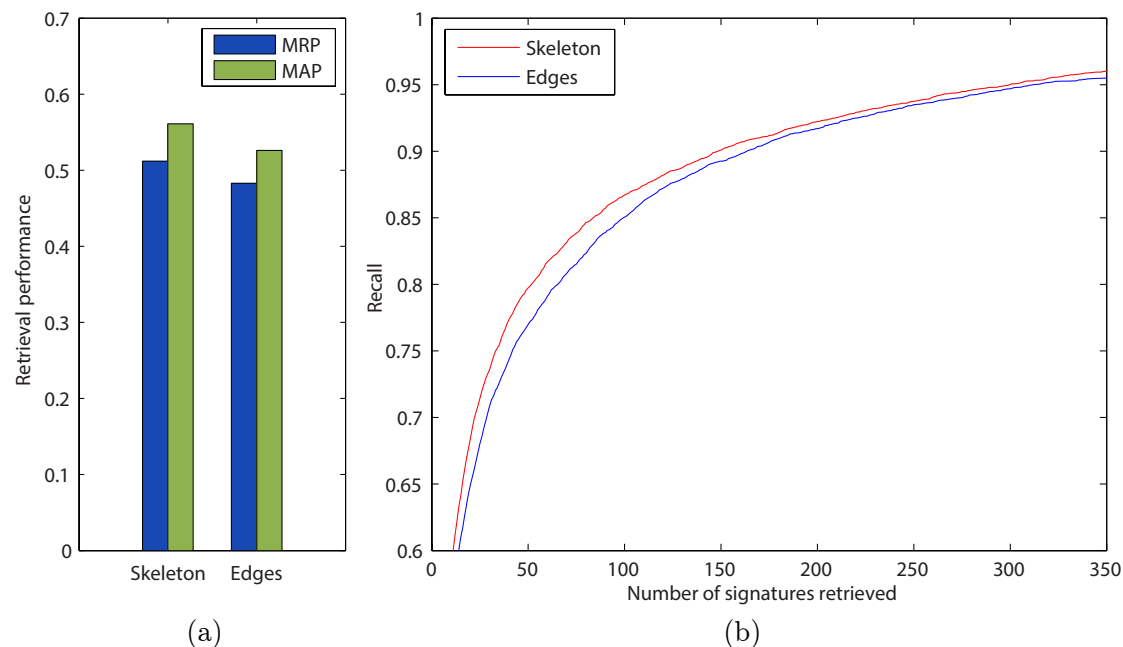


Figure 4.4: (a) The retrieval performances using the shape context distance with skeleton (left) and edge (right) images in terms of MRP and MAP. (b) The recall of the pre-filtering method using skeleton (red) and edge (blue) images with respect to the number of signature images retrieved.

Table 4.6: Retrieval performances of the shape context distance using skeleton and edge images

Method	MRP	MAP
Skeleton	51.2%	56.1%
Edges	48.3%	52.6%

The tests show that skeleton images provide better results than edge images for the shape context distance and weaker results when they are used to compute the dissimilarity measures. The superiority of edge images for the dissimilarity measures matches the observations of Zhu et al. [61]. They compare their contour-based abstraction approach, which extracts edge-like images, with two skeletonization techniques and show that their representation yields better retrieval results for the dissimilarity measures. The weaker performance of edge images for the

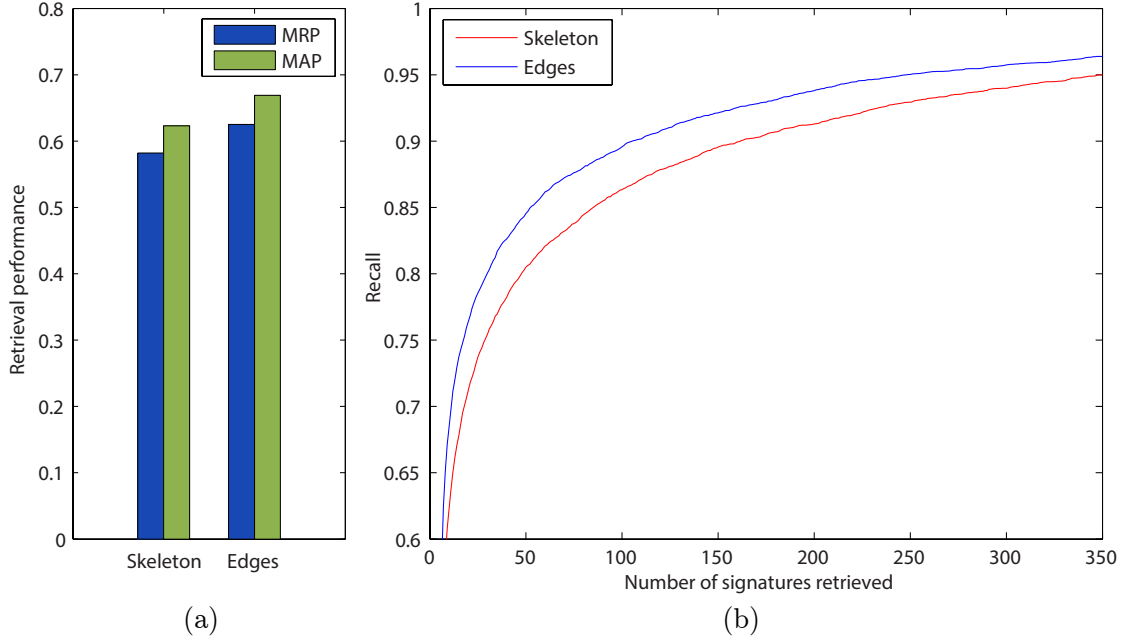


Figure 4.5: (a) The retrieval performances using the dissimilarity measures with skeleton (left) and edge (right) images in terms of MRP and MAP. (b) The recall of the dissimilarity measures using skeleton (red) and edge (blue) images with respect to the number of signature images retrieved. All results are obtained using  $weight_1$  to combine the dissimilarity measures.

Table 4.7: Retrieval performances of the dissimilarity measures using skeleton and edge images with  $weight_1$

Method	MRP	MAP
Skeleton	58.2%	62.3%
Edges	62.5%	66.9%

shape context distance, on the other hand, can be explained by the fact that edge images consist of two edges for each stroke instead of one. Since the shape context descriptor gives more importance to points in close proximity, edge images add potential for noise by having points sampled on both edges of a stroke.

## 4.4 Sampling Techniques

In this section, the effects of varying sampling rates on the performance of the signature retrieval system and its runtime are evaluated. For this purpose the

shape context distance that is used as the pre-filtering step of the signature retrieval system is computed for each image in *setB* using different sampling parameters  $q$ . Since it is shown in Section 4.3 that skeletonization yields better results when computing the shape context distance, it is the abstraction technique which is used for this test run. Figure 4.6 illustrates the results in terms of MAP and runtime in hours. The MRP is not included in the figure since it changes similarly to the MAP and would only add confusion. However, all results including the respective sampling parameters  $q$  and the average numbers of points are shown in Table 4.8.

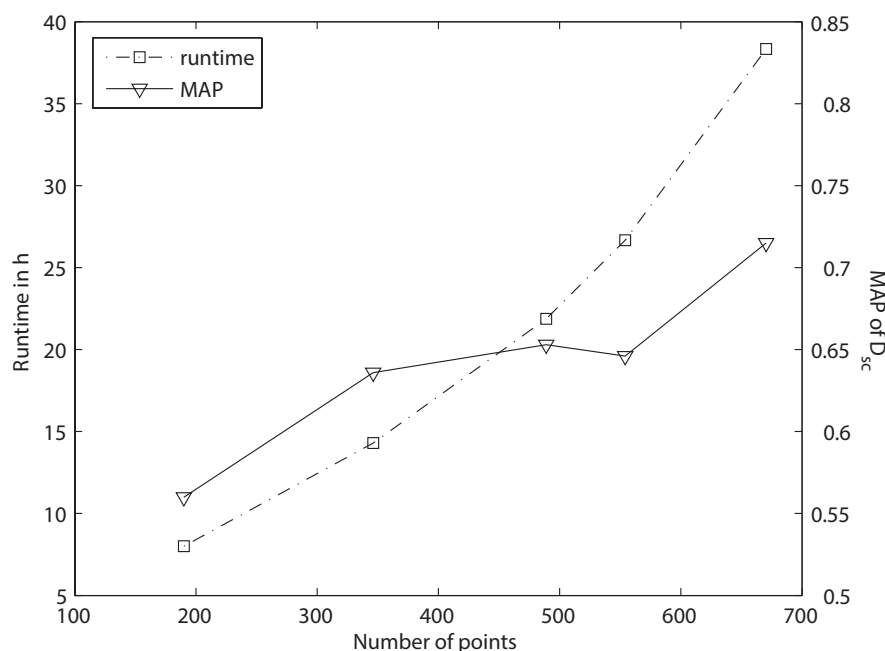


Figure 4.6: A comparison of runtime and performance in terms of MAP for different numbers of sample points using the shape context distance. The axis values corresponding to the runtime and MAP graphs are shown on the left and right side respectively.

The performance generally increases with higher numbers of sample points with an exception at  $\sim 550$  points, where the performance decreases again by 0.7 percentage points compared to the performance at  $\sim 490$  sample points. However, it can also be seen that the runtime increases exponentially with the number of points. While sampling  $\sim 350$  points increases the MAP by 7.6 percentage points compared to sampling  $\sim 190$  points, it also almost doubles the runtime. Similarly, the performance significantly improves when sampling  $\sim 670$  points compared to sampling  $\sim 350$  points, but the runtime increases simultaneously by a factor of about 2.7.

Table 4.8: Retrieval performances and runtimes using the shape context distance on *setB* with different sampling parameters

$q$	Number of points	MRP	MAP	Runtime
0.007	~190	51.2%	56.1%	7.98 h
0.015	~346	59.2%	63.7%	14.32 h
0.025	~489	60.5%	65.4%	21.87 h
0.03	~554	60.1%	64.7%	26.67 h
0.05	~670	66.6%	71.6%	38.34 h

The best trade-off between runtime and performance is therefore achieved when sampling ~350 points.

To find optimal sampling properties for the retrieval system it is also important to look at the performance of the dissimilarity measures at varying numbers of sample points. This test run is performed using edges as abstraction technique, since Section 4.3 shows that the dissimilarity measures achieve better results on edge images. The results are computed on *setA* for the sake of computation time using  $weight_2$  for combining the dissimilarity measures. They are presented in Figure 4.7 in terms of MAP and runtime. The MRP is again omitted for better clarity and all results are shown in Table 4.9.

Table 4.9: Retrieval performances and runtimes using the dissimilarity measures with  $weight_2$  on *setA* with different sampling parameters

$q$	Number of points	MRP	MAP	Runtime
0.005	~152	55.6%	62.8%	7.75 h
0.007	~208	67.1%	73.1%	17.64 h
0.009	~260	72.0%	78.1%	31.78 h
0.011	~312	75.5%	80.9%	51.89 h
0.015	~405	75.6%	81.5%	108.14 h
0.02	~516	77.8%	83.2%	215.28 h

For the dissimilarity measures it can again be observed that the performance increases with higher numbers of sample points while the runtime increases exponentially. However, the performance increase falls beneath 2 percentage points per 100 additional points, once the number of sample points reaches ~300. So while the performance in terms of MAP increases by merely 0.6 percentage points when sampling ~410 points instead of ~310, the runtime doubles to more than 108 hours on the small *setA*. It is therefore not reasonable to sample more than ~300 points, because the performance increase does not justify the enormous runtime. It is also worth noting that the evaluation with ~310 points takes more than 2 days on a

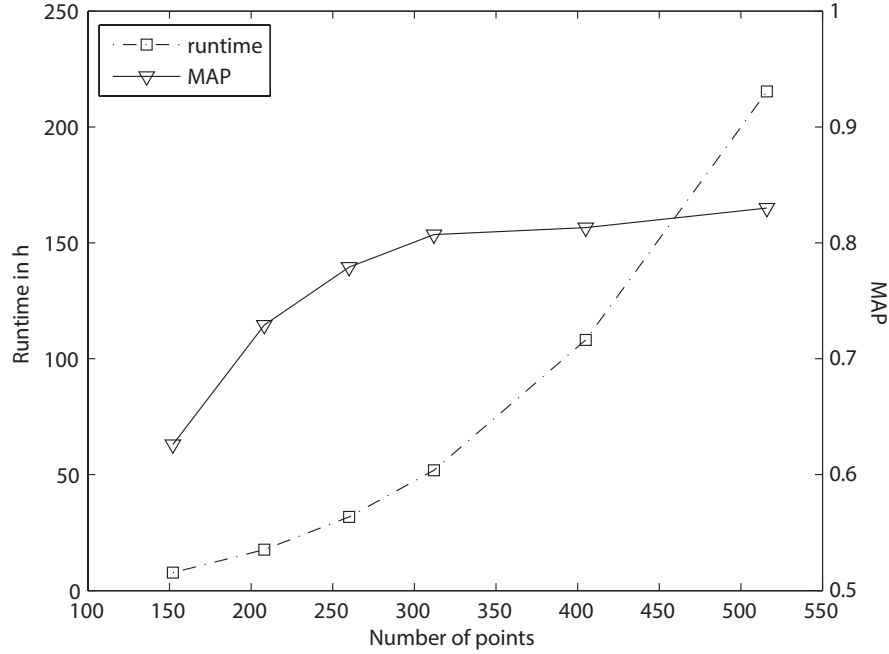


Figure 4.7: A comparison of runtime and performance in terms of MAP for different numbers of sample points using the dissimilarity measures with  $weight_2$ . The axis values corresponding to the runtime and MAP graphs are shown on the left and right side respectively.

set with only 200 signature images. For larger sets it is therefore recommended to sample less than 300 points. Hence, the number of points sampled for computing the dissimilarity measures in all other sections of this chapter is  $\sim 200$  which is also the number of points that Zhu et al. [61] use for their evaluation.

## 4.5 Hybrid Approach

Since it is shown in Section 4.3 that the pre-filtering step performs better on skeleton images and the dissimilarity measures achieve better results on edge images, the retrieval system that is proposed in this thesis uses a hybrid approach. Due to the insights obtained in Section 4.4 regarding the optimal number of sample points, the shape context distance in the pre-filtering step is computed by sampling about 350 points (i.e.  $q = 0.015$ ) on skeleton images and the dissimilarity measures for the reduced set are then computed by sampling about 200 points (i.e.  $q = 0.007$ ) on edge images. The following subsections evaluate the different aspects of the hybrid approach, namely the size of the reduced set, the performance compared to

the dissimilarity measures, the effects of using training data and the performance of single distances.

#### 4.5.1 Reduced Set Size

As mentioned in Section 3.5 the shape context distance is used for pre-filtering such that the dissimilarity measures are computed for only up to five percent of the signature images in the set. In order to see differences in varying sizes of this reduced set, the signature retrieval system is evaluated on *setB* with  $weight_2$  using the pre-filtering step with reduced sets from one to five percent. Figure 4.8 illustrates the results in terms of MRP and MAP while the exact values are shown in Table 4.10. The results show that – just as one would expect – the performance increases with the size of the reduced set. However, while the step from 1% to 2% reduced set size yields a significant performance boost of 1.9 percentage points MRP and 1.4 percentage points MAP all further steps yield only small improvements of 0.3 percentage points or less. Note that the last step from 4% to 5% results in an increased retrieval performance of merely 0.1 percentage points. These results suggest that the best trade-off between retrieval performance and runtime is achieved with a reduced set of about 3%.

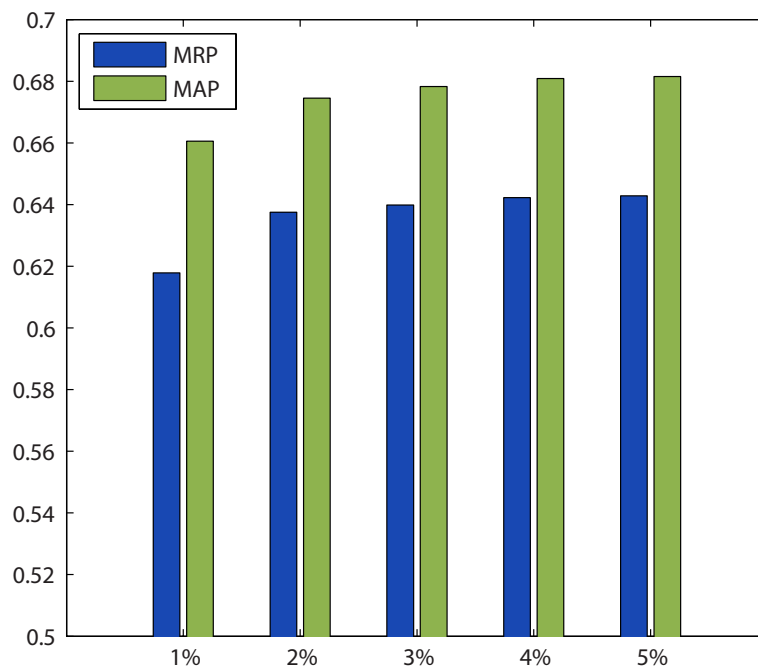


Figure 4.8: Retrieval performances in terms of MRP and MAP with different reduced set sizes.

Table 4.10: Retrieval performances using pre-filtered sets of different sizes. The exact number of signatures in the reduced set is shown in parentheses

Performance	1% (10)	2% (19)	3% (29)	4% (38)	5% (48)
MRP	61.8%	63.7%	64.0%	64.2%	64.3%
MAP	66.1%	67.5%	67.8%	68.1%	68.2%
Runtime	18.57 h	21.86 h	23.80 h	26.28 h	30.57 h

#### 4.5.2 Comparison with Zhu et al.

Since Zhu et al. [61] evaluate the dissimilarity measures on a different dataset, namely the *Tobacco-800* [32] set which consists of real world documents from US tobacco companies, their results cannot directly be compared to the results in this thesis. For this reason both the dissimilarity measures on their own and the hybrid approach using the dissimilarity measures with the pre-filtering step are evaluated on *setB* to see how they perform in comparison. Regarding the size of the dataset used by Zhu et al. they state that *Tobacco-800* contains 66 classes with 6-11 signatures each, which results in 396-726 signatures in total. Since 20% are used as training data this leaves 317-581 signatures that are left as test data. The test set used in their evaluation is therefore smaller than *setB*.

The results in terms of MRP and MAP are visualized in Figure 4.9 (a) and a comparison of the recall of both methods at each rank is given in Figure 4.9 (b). The exact values including the total runtime of the experiments are shown in Table 4.11 and the weight used for combining the four dissimilarity measures is  $weight_2$ .

Table 4.11: Retrieval performances and runtimes on *setB* using  $weight_2$

Method	MRP	MAP	Runtime
Dissimilarity measures	62.4%	66.9%	16.71 days
Hybrid approach (3%)	64.0%	67.8%	0.99 days

The results show that the hybrid approach with a reduced set of 3% provides a speed-up of factor 16 on the test set and even achieves slightly better retrieval results in terms of MRP and MAP than the dissimilarity measures on their own. It can be seen in Figure 4.9 (b), however, that the hybrid approach has a lower recall rate when about 20-80 signatures are retrieved which means that the dissimilarity measures are more likely to rank relevant signatures at these positions than the hybrid approach. This effect occurs due to the reduced set which contains only 29 signatures in this case and can be reduced by increasing its size. Note that the hybrid approach with a reduced set of 5% still provides a speed-up of factor 13 and

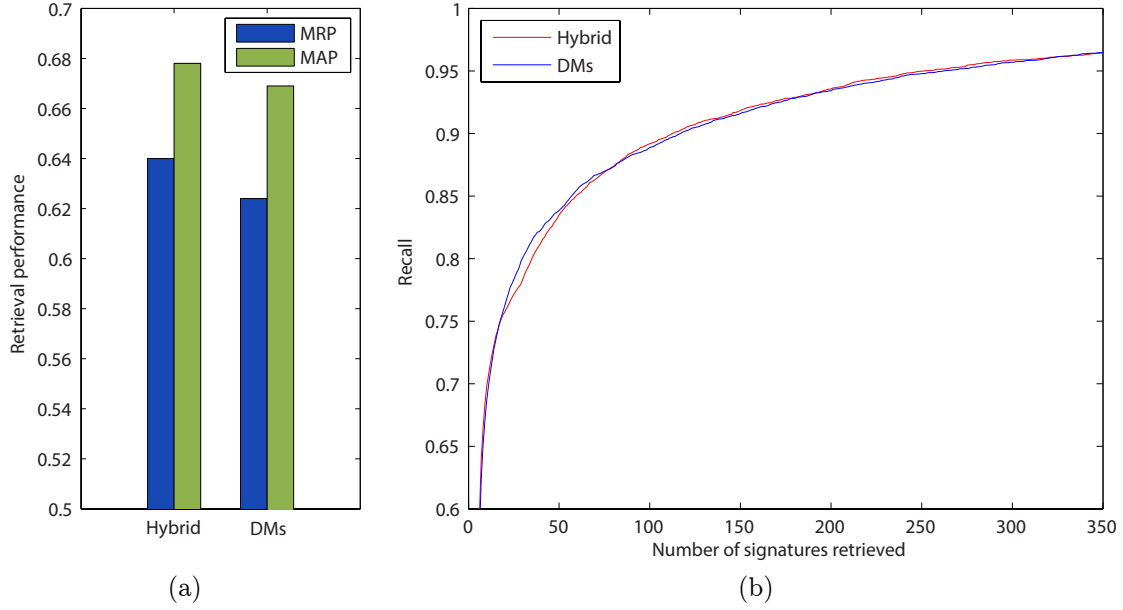


Figure 4.9: The results (a) in terms of MRP and MAP and (b) the average recall of the hybrid approach with a reduced set size of 3% (red) and the dissimilarity measures (blue).

achieves a 1.9 percentage points higher MRP and a 1.3 percentage points higher MAP compared to the dissimilarity measures (see Table 4.10).

### 4.5.3 Training Data

It is mentioned in Section 3.4 that the dissimilarity measures can also be combined without using pre-computed weights, thus making it possible to retrieve signature images without the need for training data. This section evaluates the impact of using training data on the retrieval performance. For this purpose the hybrid approach with a reduced set of 3% and the dissimilarity measures are both evaluated on *setB* using  $weight_1$ ,  $weight_2$  and no weights. The results of this test in terms of MRP and MAP are shown in Figure 4.10 and Table 4.12. It can be seen that the hybrid approach achieves the best results with  $weight_2$ , while  $weight_1$  yields the worst performance of the three. The results also show that it is possible to obtain only slightly lower performance rates without using any training data than with  $weight_2$  which uses 25% training data. To be precise, the MRP and MAP of the hybrid approach with  $weight_2$  are only 0.3 and 0.1 percentage points higher than without weights.

The results of the dissimilarity measures show even better performance without



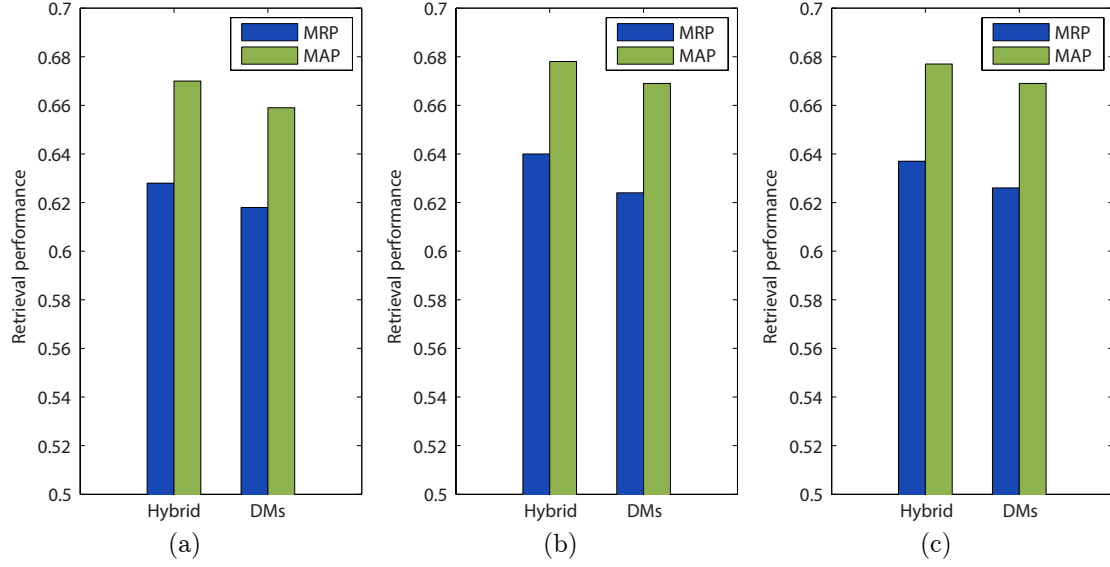


Figure 4.10: The retrieval performances in terms of MRP and MAP using (a)  $weight_1$ , (b)  $weight_2$  and (c) no weights for the hybrid approach with a reduced set of 3% and the dissimilarity measures.

Table 4.12: Retrieval performances using different weights and no weights for the hybrid approach with a reduced set of 3% and the dissimilarity measures

Performance	$weight_1$		$weight_2$		without weights	
	Hybrid	DMs	Hybrid	DMs	Hybrid	DMs
MRP	62.8%	61.8%	64.0%	62.4%	63.7%	62.6%
MAP	67.0%	65.9%	67.8%	66.9%	67.7%	66.9%

using weights than the hybrid approach. They achieve a 0.2 percentage points higher MRP and the same MAP without training data as with  $weight_2$  and a more than 0.8 percentage points higher MRP and MAP compared to using  $weight_1$ . These results suggest that it is not mandatory for the dissimilarity measures and the hybrid approach to use training data since it reduces the size of the test set. However, the *GPDS960signature* database is several times larger than *setB* which means that enough training data is available. The results in this thesis are therefore computed using weights.

#### 4.5.4 Single Distances

This section gives an overview of the performance of single distances similar to Zhu et al. [61]. The results for the dissimilarity measures and the hybrid approach

using single distances on their own are presented in Figure 4.11 and Table 4.13. Firstly it can be seen that the order in terms of retrieval performance is different for the two approaches. While for the dissimilarity measures the shape context distance ( $D_{sc}$ ) performs best, followed by the registration residual error ( $D_{re}$ ), the bending energy ( $D_{be}$ ) and the anisotropic scaling ( $D_{as}$ ), it is  $D_{re}$  which performs best for the hybrid approach followed by  $D_{be}$ ,  $D_{sc}$  and  $D_{as}$ . The only similarity here is that  $D_{as}$  performs worst for both approaches. Comparing the results of the dissimilarity measures to those of Zhu et al., it is also worth noting that  $D_{re}$  and  $D_{as}$  swapped their position. The results in Section 4.1 suggest that this is due to the performance gain from using the weighted registration residual error implementation ( $D_{re}^W$ ).

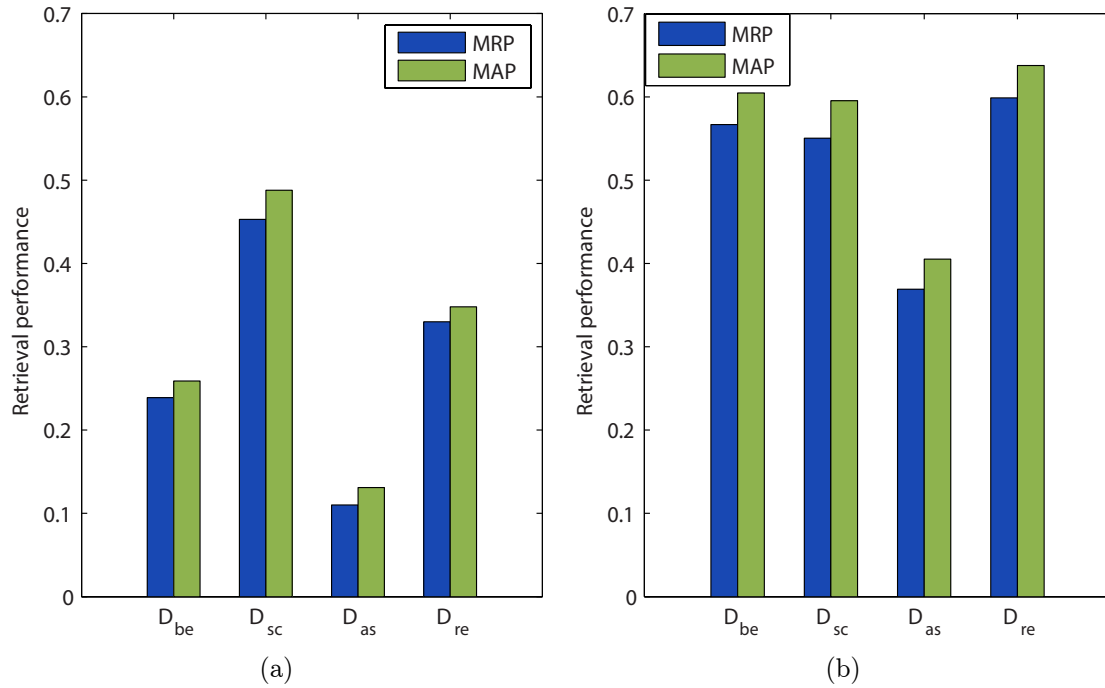


Figure 4.11: The retrieval performance of single distances in terms of MRP and MAP for (a) the dissimilarity measures and (b) the hybrid approach with a reduced set of 3%.

Secondly the results show that the retrieval performance for single distances is significantly higher (i.e. up to 34.6 percentage points for  $D_{be}$ ) for the hybrid approach than for the dissimilarity measures. This can be explained by the fact that each distance profits from the pre-filtering step used in the hybrid approach, thus resulting in a better retrieval performance for each distance on its own.

Table 4.13: Retrieval performances of single distances for the dissimilarity measures and the hybrid approach with a reduced set of 3%

Distance	Dissimilarity measures		Hybrid approach (3%)	
	MRP	MAP	MRP	MAP
$D_{be}$	23.9%	25.9%	56.7%	60.5%
$D_{sc}$	45.3%	48.8%	55.0%	59.5%
$D_{as}$	11.0%	13.1%	36.9%	40.5%
$D_{re}$	33.0%	34.8%	59.9%	63.8%

## 4.6 Prototype

In the course of the work on this thesis a prototype was developed which allows to use the methods presented for signature image retrieval. The Graphical User Interface (GUI) of the prototype is shown in Figure 4.12. It allows to choose a signature image to retrieve similar images from a signature library. The weights for combining the dissimilarity measures can be selected from the menu or imported from a custom file which contains the weights. Alternatively it is possible not to use weights but instead compute the overall distance as described in Section 3.4. Depending on the method selected on the left, different approaches are used for searching the library. The *Fast* method scans the library using only shape context distances, while the *Thorough* method computes the dissimilarity measures for all images in the library. The *Normal* method corresponds to the hybrid approach presented in this thesis using the shape context distance for pre-filtering and computing the dissimilarity measures for the reduced set only. The exact workflow of each method is described in Table 4.14. Once the search is completed the images in the library are ranked according to their distance and presented in the *Results* window (see Figure 4.13) where each image is shown with its file name and the distance from the query image.

## Summary

In this chapter the key aspects of the signature retrieval system presented in this thesis were evaluated using two subsets of the *GPDS960signature* database. First the three different implementations of the registration residual error (i.e.  $D_{re}^H$ ,  $D_{re}^D$  and  $D_{re}^W$ ) were evaluated. The tests showed that the two implementations presented in this thesis perform significantly better than  $D_{re}^H$ , both in terms of retrieval performance and runtime. The BFE was then compared to a shape-context-based and a gradient orientation normalization technique to evaluate the quality of the orientation normalization of the retrieval system. The experiments



Figure 4.12: The main window of the prototype which allows to search signature image libraries with a query signature. The numbers on the left indicate the steps necessary in order to search a signature library.

showed that the best retrieval performance is achieved using the BFE.

In the following sections the performance of the pre-filtering step and the dissimilarity measures was evaluated using different abstraction techniques, namely skeletonization and edge detection, and varying numbers of sample points. The results showed that the pre-filtering step performs best (in terms of retrieval performance and runtime) when sampling about 350 points on skeleton images and that the dissimilarity measures achieve the best results when sampling about 200 points on edge images. This led to the proposal of the hybrid approach which uses skeleton images for the pre-filtering step and edge images for computing the dissimilarity measures on the reduced set.

Subsequently it was demonstrated that the best trade-off between retrieval performance and runtime is obtained by using a reduced set of 3%. The performance of the hybrid approach with a reduced set of 3% was then compared to that of the dissimilarity measures on *setB*. This test showed that the hybrid approach yields a speed-up of factor 16 while still achieving slightly better performance results com-

Table 4.14: Workflow for the three available methods

Fast	Normal	Thorough	
Scan library for image files			
Preprocess signature images			
Compute shape context descriptors for all images			
Compute shape context distances to query image			
Sort results			
	Select reduced set		
	Compute TPS transformations		
	Compute dissimilarity measures		
	Combine dissimilarity measures and sort results		
Display results			

pared to the dissimilarity measures. The next test analysed the impact of using training data and led to the conclusion that it is not mandatory to use training data for the signature retrieval system proposed in this thesis. The performance of each single distance on its own was evaluated in the following section and demonstrated significant differences between the hybrid approach and the dissimilarity measures. Finally, the prototype which was developed in the course of this thesis was presented together with the workflow of the available options for searching signature libraries.

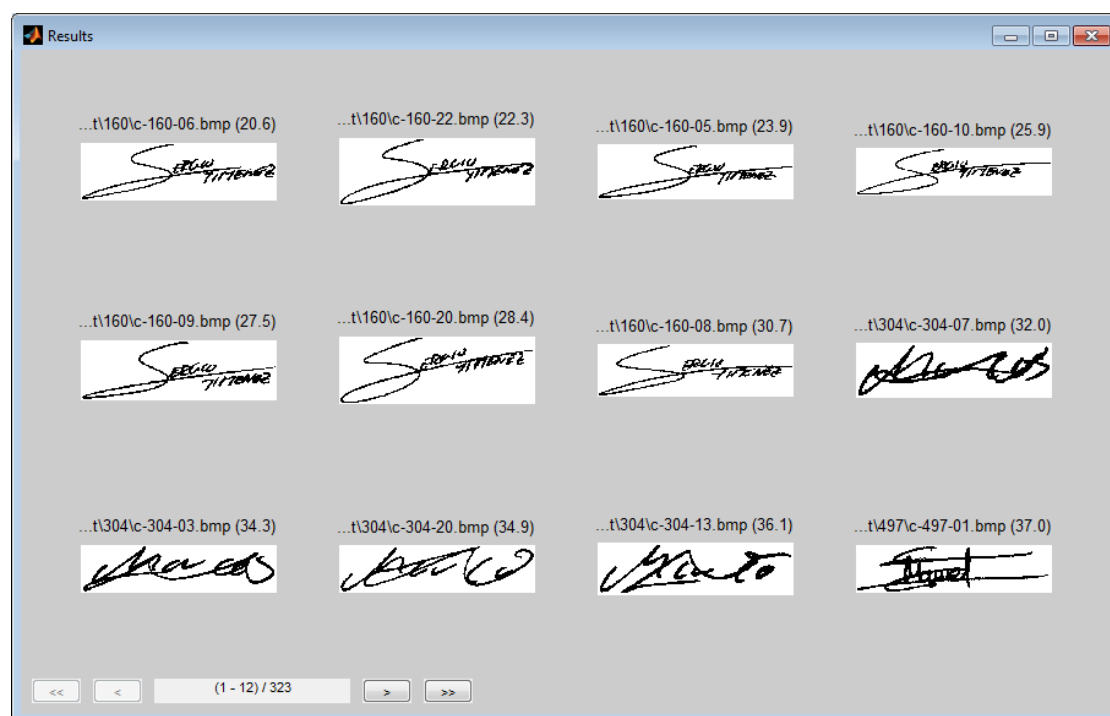


Figure 4.13: The results window which allows the user to browse the signature images ranked by their distance (shown in parentheses).

## Conclusion

In this thesis a hybrid approach is proposed that combines a state-of-the-art document image retrieval method with a pre-filtering step. The images are first preprocessed by normalizing their orientation using the BFE and their size in terms of diagonal length. As demonstrated in Section 3.1.2 the orientation normalization is necessary in order for the shape-context-based pre-filtering step to work properly. The pre-filtering step subsequently reduces the test set by computing the shape context distance between the query signature and each signature in the dataset and ranking the signatures accordingly. The distances to the top 3% of signatures are then re-computed using the TPS-RPM algorithm and computing four dissimilarity measures from the resulting TPS transformation. Finally, all the signature images are ranked by their combined distance.

The signature retrieval system proposed in this thesis was evaluated on two subsets of the *GPDS960signature* database. First, the performance improvements of the two registration residual error implementations proposed in this thesis were demonstrated by comparing them to the results of the registration residual error implementation which uses the hungarian method for finding optimal matches. It was then shown that the BFE yields the best retrieval performance compared to two other orientation normalization techniques, namely a shape-context-based normalization and a gradient orientation normalization.

Subsequent experiments analysed the effects of using skeletonization or edge detection as abstraction technique. It was observed that the dissimilarity measures yield better results using edge images while the shape context distance in the pre-filtering step performed better on skeleton images. This was argued to be due to the fact that edge images comprise stroke thickness information which adds noise to the shape context descriptor.

Further experiments were conducted to find the best trade-off between performance and runtime for varying numbers of sample points. The results of these experiments led to the proposal of the hybrid approach which performs the pre-filtering step on skeleton images using a higher sampling rate while the dissimilarity measures are computed on edge images with a lower sampling rate. The analysis of the size of the reduced set demonstrated that a reduced set of 3% yields the best trade-off between performance and runtime. It was shown that the pre-filtering brings a significant speed-up while providing slightly better retrieval results than the dissimilarity measures on their own. The reason why the shape context distance is used to estimate correspondences is that after the normalization of the images in the preprocessing step similar signatures have a low shape context distance even without knowing the transformation between them.

Additional evaluations demonstrated that the use of training data has only a small effect on the retrieval performance which means that it is not mandatory to train the signature retrieval system. Finally, the comparison of the performance of single distance measures showed that each distance measure benefits from the pre-filtering step in the hybrid approach, thus achieving significantly better results than without the pre-filtering step.

## 5.1 Disadvantages

The most obvious disadvantage of the signature retrieval system presented in this thesis is its runtime. Since the dissimilarity measures are computed using the transformation from one signature to another, it is necessary to run the computationally expensive TPS-RPM algorithm for each image in a signature library in order to retrieve images. Additionally, the transformation cannot be pre-computed for the signature library since each new query image has different transformations to the images in the library. This is because the retrieval system is based on distances instead of feature vectors which can be computed once for the entire signature library.

## 5.2 Advantages

As presented in Section 4.5 the hybrid approach proposed in this thesis is more than 16 times faster than the dissimilarity measures on their own. Moreover it also achieves slightly better retrieval results in terms of MRP and MAP. The signature retrieval system therefore clearly benefits from the pre-filtering step using the shape context distance. It was also shown in Section 4.5.3 that the performance of the retrieval system does not rely on the use of training data. It can therefore be used



with different signature image libraries without the need for training it with every new library.

### 5.3 Future Work

Since this work solely focuses on the matching and retrieval of signature images, the system could be extended by a localization algorithm which detects signatures in document images. Furthermore, additional preprocessing elements such as printed text removal and filtering of noise could be added. The system can also be improved by implementing a more robust orientation normalization technique which would further improve the performance of the pre-filtering step. If the system is extended to document image retrieval by adding a signature localization it is also recommendable to improve the TPS-RPM algorithm to support outlier handling in both point sets as proposed by [58] since real world documents contain more noise than the binarized signature images contained in the *GPDS960signature* database.

Future work may also try to adapt an approach for writer retrieval presented by Fiel and Sablatnig [18]. They use a bag of words approach with SIFT features to retrieve documents from a specific writer. The documents are ranked using the  $\chi^2$  distance between histograms of the cluster centres of the SIFT features. The advantage of this approach is that it is based on features which can be pre-computed and only have to be converted to distances for retrieval.

# Bibliography

- [1] G. Agam and S. Suresh. Warping-Based Offline Signature Recognition. *IEEE Transactions on Information Forensics and Security*, 2(3):430–437, September 2007.
- [2] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
- [3] Paul J Besl and Neil D McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [4] M. Blumenstein, M. A. Ferrer, and J. F. Vargas. The 4NSigComp2010 Off-line Signature Verification Competition: Scenario 2. In *12th International Conference on Frontiers in Handwriting Recognition*, pages 721–726. IEEE, November 2010.
- [5] F.L. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.
- [6] 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, New York, New York, USA, July 2000. ACM Press.
- [7] Wilhelm Burger and Mark James Burge. *Principles of Digital Image Processing - Core Algorithms*. Undergraduate Topics in Computer Science. Springer, 2009.
- [8] J.L. Camino, C.M. Travieso, C.R. Morales, and M.A. Ferrer. Signature Classification by Hidden Markov Model. In *International Carnahan Conference on Security Technology*, pages 481–484. IEEE, 1999.

- [9] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.
- [10] H. Chui and A. Rangarajan. A New Point Matching Algorithm for Non-Rigid Registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, February 2003.
- [11] C Cortes and V Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.
- [12] F. Davoine, M. Antonini, J.-M. Chassery, and M. Barlaud. Fractal Image Compression Based on Delaunay Triangulation and Vector Quantization. *IEEE Transactions on Image Processing*, 5(2):338–346, 1996.
- [13] Markus Diem, Florian Kleber, and Robert Sablatnig. Text Classification and Document Layout Analysis of Paper Fragments. In *International Conference on Document Analysis and Recognition*, pages 854–858. IEEE, September 2011.
- [14] Markus Diem and Robert Sablatnig. Recognizing Degraded Handwritten Characters. Technical report, Computer Vision Lab, 2011.
- [15] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, 2nd edition, 2001.
- [16] José L. Esteban, José F. Vélez, and Ángel Sánchez. Off-line Handwritten Signature Detection by Analysis of Evidence Accumulation. *International Journal on Document Analysis and Recognition*, 15(4):359–368, October 2011.
- [17] JT Favata, G Srikantan, and SN Srihari. Handprinted Character/Digit Recognition using a Multiple Feature/Resolution Philosophy. In *Frontiers in Handwriting Recognition*, pages 57–66, 1994.
- [18] Stefan Fiel and Robert Sablatnig. Writer Retrieval and Writer Identification Using Local Features. In *IAPR International Workshop on Document Analysis Systems*, pages 145–149. IEEE, March 2012.
- [19] RA Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179–188, 1936.
- [20] Angelika Garz, Robert Sablatnig, and Markus Diem. Layout Analysis for Historical Manuscripts Using Sift Features. In *International Conference on Document Analysis and Recognition*, pages 508–512. IEEE, September 2011.

- [21] Ke Han and Ishwar K. Sethi. Handwritten Signature Retrieval and Identification. *Pattern Recognition Letters*, 17(1):83–90, January 1996.
- [22] MK Hu. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.
- [23] Cheolkon Jung, Qifeng Liu, and Joongkyu Kim. Accurate Text Localization in Images Based on SVM Output Scores. *Image and Vision Computing*, 27(9):1295–1301, August 2009.
- [24] Keechul Jung, Kwang In Kim, and Anil K. Jain. Text Information Extraction in Images and Video: A Survey. *Pattern Recognition*, 37(5):977–997, May 2004.
- [25] Edson J.R. Justino, Flávio Bortolozzi, and Robert Sabourin. A Comparison of SVM and HMM Classifiers in the Off-line Signature Verification. *Pattern Recognition Letters*, 26(9):1377–1385, July 2005.
- [26] Maya V. Karki, K. Indira, and S. Sethu Selvi. Off-line Signature Recognition and Verification Using Neural Network. In *International Conference on Computational Intelligence and Multimedia Applications*, volume 01, pages 307–312. IEEE, December 2007.
- [27] A. Karouni, B. Daya, and S. Bahlak. Offline Signature Recognition Using Neural Networks Approach. *Procedia Computer Science*, 3:155–161, January 2011.
- [28] E Kavallieratou and DC Balcan. Handwritten Text Localization in Skewed Documents. In *International Conference on Image Processing*, pages 1102–1105, 2001.
- [29] Florian Kleber, Markus Diem, and Robert Sablatnig. Scale Space Binarization Using Edge Information Weighted by a Foreground Estimation. *International Conference on Document Analysis and Recognition*, pages 1180–1184, September 2011.
- [30] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, March 1955.
- [31] K Kuhnke, L Simoncini, and Zs M Kovács-V. A System for Machine-Written and Hand-Written Character Distinction. In *Document Analysis and Recognition*, pages 811–814, 1995.

- [32] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. Building a Test Collection for Complex Document Information Processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 665, New York, New York, USA, August 2006. ACM Press.
- [33] C.C. Lin and C.T. Chang. A Fast Shape Context Matching Using Indexing. In *International Conference on Genetic and Evolutionary Computing*, pages 17–20. IEEE, August 2011.
- [34] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [35] Ines Ben Messaoud, Hamid Amiri, Haikal El Abed, and Volker Margner. New Binarization Approach Based on Text Block Extraction. *International Conference on Document Analysis and Recognition*, pages 1205–1209, September 2011.
- [36] Atul Negi and Nikhil Kasinadhuni. Localization and Extraction of Text in Telugu Document Images. In *Conference on Convergent Technologies for the Asia-Pacific Region*, pages 749–752, 2003.
- [37] Xiao-Xiao Niu and Ching Y. Suen. A Novel Hybrid CNN–SVM Classifier for Recognizing Handwritten Digits. *Pattern Recognition*, 45(4):1318–1325, April 2012.
- [38] N Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, (1):62–66, 1979.
- [39] Cemil Oz. Signature Recognition and Verification with Artificial Neural Network Using Moment Invariant Method. In *International Symposium on Neural Networks*, volume 3497 of *Lecture Notes in Computer Science*, pages 195–202. Springer, 2005.
- [40] I. Pavlidis, N.P. Papanikolopoulos, and R. Mavuduru. Signature Identification Through the Use of Deformable Structures. *Signal Processing*, 71(2):187–201, December 1998.
- [41] Ofir Pele and Michael Werman. The Quadratic-Chi Histogram Distance Family. In *European Conference on Computer Vision*, number 1, pages 1–14, 2010.
- [42] R. Plamondon and Sargur N Srihari. Online and Off-line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

- [43] L Rabiner and BH Juang. An Introduction to Hidden Markov Models. *ASSP Magazine*, 3(1):4–16, June 1986.
- [44] Kandan Ramakrishnan, KR Arvind, and AG Ramakrishnan. Localization of Handwritten Text in Documents Using Moment Invariants and Delaunay Triangulation. In *International Conference on Computational Intelligence and Multimedia Applications*, pages 408–414. IEEE, December 2007.
- [45] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The Earth Mover’s Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
- [46] J Sauvola and M Pietikäinen. Adaptive Document Image Binarization. *Pattern Recognition*, 33(2):225–236, 2000.
- [47] J. Schneider and B. Nickolay. The Stasi Puzzle. *Fraunhofer Magazine, Special Issue*, 1:32–33, 2008.
- [48] M. S. Shirdhonkar and Manesh B. Kokare. Document Image Retrieval Using Signature as Query. In *International Conference on Computer and Communication Technology*, pages 66–70. IEEE, September 2011.
- [49] M. S. Shirdhonkar and Manesh B. Kokare. Off-line Handwritten Signature Identification Using Rotated Complex Wavelet Filters. *Journal of Computer Science*, 8(1):478–482, 2011.
- [50] Sargur N Srihari, Shravya Shetty, Siyuan Chen, Harish Srinivasan, Chen Huang, G. Agam, and O. Frieder. Document Image Retrieval Using Signatures as Queries. In *International Conference on Document Image Analysis for Libraries*, pages 198–203. IEEE, 2006.
- [51] Bolan Su, Shijian Lu, and Chew Lim Tan. Binarization of Historical Document Images Using the Local Maximum and Minimum. In *International Workshop on Document Analysis Systems*, pages 159–165, 2010.
- [52] YY Tang, SW Lee, and CY Suen. Automatic Document Processing: A Survey. *Pattern Recognition*, 29(12):1931–1952, 1996.
- [53] MR Teague. Image Analysis via the General Theory of Moments. *Journal of the Optical Society of America*, 70(8):920–930, 1980.
- [54] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*, volume 50. SIAM, 1997.

- [55] RE Uhrig. Introduction to Artificial Neural Networks. In *International Conference on Industrial Electronics, Control, and Instrumentation*, volume 19, pages 33–37, December 1995.
- [56] J. Van Beusekom, D. Keysers, F. Shafait, and T.M. Breuel. Distance Measures for Layout-Based Document Image Retrieval. In *International Conference on Document Image Analysis for Libraries*, pages 232–242. IEEE, 2006.
- [57] J.F. Vargas, M.A. Ferrer, C.M. Travieso, and J.B. Alonso. Off-line Signature Verification Based on Grey Level Information Using Texture Features. *Pattern Recognition*, 44(2):375–385, February 2011.
- [58] Jinzhong Yang. The Thin Plate Spline Robust Point Matching (TPS-RPM) Algorithm: A Revisit. *Pattern Recognition Letters*, 32(7):910–918, May 2011.
- [59] TY Zhang and CY Suen. A Fast Parallel Algorithm for Thinning Digital Patterns. *Image Processing and Computer Vision*, 27(3):236–239, 1984.
- [60] Xiaowei Zheng, Wei Tan, and Jianhong Du. A Fast Adaptive Binarization Method Based on Sub Block OSTU and Improved Sauvola. In *International Conference on Wireless Communications, Networking and Mobile Computing*, number 1, pages 1–5. IEEE, September 2011.
- [61] G. Zhu, Y. Zheng, D. Doermann, and S. Jaeger. Signature Detection and Matching for Document Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2015–2031, November 2009.

# List of Acronyms

<b>AP</b>	Average Precision
<b>BB</b>	Bounding Box
<b>BFE</b>	Best-Fit Ellipse
<b>CC</b>	Connected Component
<b>DT-CWT</b>	Dual Tree Complex Wavelet Transform
<b>DT-RCWF</b>	Dual Tree Rotated Complex Wavelet Filter
<b>DTW</b>	Dynamic Time Warping
<b>EMD</b>	Earth Mover's Distance
<b>FFT</b>	Fast Fourier Transform
<b>GDR</b>	German Democratic Republic
<b>GSC</b>	Gradient, Structural and Concavity
<b>GSF</b>	Gradient Shape Feature
<b>GUI</b>	Graphical User Interface
<b>HMM</b>	Hidden Markov Model
<b>ICP</b>	Iterative Closest Point
<b>LCS</b>	Longest Common Subsequence
<b>LDA</b>	Linear Discriminant Analysis
<b>MAP</b>	Mean Average Precision
<b>MRP</b>	Mean R-Precision



<b>NN</b>	Neural Network
<b>PCA</b>	Principal Component Analysis
<b>RBF</b>	Radial Basis Function
<b>RP</b>	R-Precision
<b>SIFT</b>	Scale-Invariant Feature Transform
<b>SVM</b>	Support Vector Machine
<b>TPS</b>	Thin-Plate Spline
<b>TPS–RPM</b>	Thin-Plate Spline – Robust Point Matching

