A Generic Framework for Information Segmentation in Document Images: A Part-based Approach

Dissertation

approved by the

Department of Computer Science
Technische Universität Kaiserslautern
for the fulfillment of the requirements for the Doctoral Degree

Doctor of Engineering (Dr. Ing)

by

Sheraz Ahmed

Date of Viva	**:**	15 December 2015
Dean	**:**	Prof. Dr. Klaus Schneider
PhD Committee Chairperson	**:**	Prof. Dr. Sebastian Michel
Reviewers	**:**	Prof. Dr. Prof. h.c. Andreas Dengel apl. Prof. Dr. habil. Marcus Liwicki

D 386

Sheraz Ahmed
Wilhelm-Raabe-Strae 26,
67663 Kaiserslautern

Kaiserslautern, December 01, 2016

Erklärung

Ich versichere hiermit, dass ich die vorliegende Promotionarbeit mit dem Thema "*A Generic Framework For Information Segmentation in Document Images: A part-based approach*" selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, habe ich durch die Angabe der Quelle, auch der benutzten Sekundärliteratur, als Entlehnung kenntlich gemacht.

(Sheraz Ahmed)

To my dearest parents (Mera Pyara Maa Jee and Mera Sohna Abu Jee).

Acknowledgement

First of all, my greatest gratitude to the Almighty who has enabled me complete the thesis at hand and made this day possible for me to write this acknowledgment.

A special thanks to Prof. Dr. Prof. h.c. Andreas Dengel, I am really honored to have an opportunity to work in your group to complete this thesis. This has been one of the best experiences of my life. You have always encouraged and motivated me. Surely, you are a great source of inspiration for young researchers. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would like to thank apl.-Prof. Dr. habil. Marcus Liwicki, who has always remained a firm support for me during this entire study endeavor. Thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career have been priceless and a great source of motivation for me.

I would like to thank my best friend, Imran. His continuous motivation and support has always been my asset. His on and off discussions helped me a lot in understanding various issues concerned with my life, my studies, and me. Special thanks to my friends, Mohsin, Adeel, Zain, Sajid, Abdullah, and Gulzar. Their company and support has helped greatly throughout my PhD.

A special thanks to all of my colleagues at the KM group, DFKI Kaiserslautern. I would also like to say thanks to Prof. Koichi Kise, Prof. Masakazu Iwamura, Prof. Seiichi Uchida, Prof. Ajmal Mian, and Prof. Faisal Shafait, who gave me opportunity to visit their labs (in Japan and Australia) which helped me a lot in getting experience of working with international colleagues and in broaden my vision.

In the last but not the least, I really really admire the support of my family: my parents, my wife, my brothers, and my sister. Words cannot express how grateful I am to all of you for all of the sacrifices that you have made on my behalf. Your prayer and support for me was what sustained me thus far. Without this support, it was never possible for me to think of this day. A huge bouquet of love, thanks, and gratitude for my family.

Executive Summary

This thesis presents a novel, generic framework for information segmentation in document images. A document image contains different types of information, for instance, text (machine printed/handwritten), graphics, signatures, and stamps. It is necessary to segment information in documents so that to process such segmented information only when required in automatic document processing workflows.

The main contribution of this thesis is the conceptualization and implementation of an information segmentation framework that is based on part-based features. The generic nature of the presented framework makes it applicable to a variety of documents (technical drawings, magazines, administrative, scientific, and academic documents) digitized using different methods (scanners, RGB cameras, and hyper-spectral imaging (HSI) devices). A highlight of the presented framework is that it does not require large training sets, rather a few training samples (for instance, four pages) lead to high performance, i.e., better than previously existing methods. In addition, the presented framework is simple and can be adapted quickly to new problem domains. This thesis is divided into three major parts on the basis of document digitization method (scanned, hyper-spectral imaging, and camera captured) used.

In the area of scanned document images, three specific contributions have been realized. The first of them is in the domain of signature segmentation in administrative documents. In some workflows, it is very important to check the document authenticity before processing the actual content. This can be done based on the available seal of authenticity, e.g., signatures. However, signature verification systems expect pre-segmented signature image, while signatures are usually a part of document. To use signature verification systems on document images, it is necessary to first segment signatures in documents. This thesis shows that the presented framework can be used to segment signatures in administrative documents. The system based on the presented framework is tested on a publicly available dataset where it outperforms the state-of-the-art methods and successfully segmented all signatures, while less than half of the found signatures are false positives. This shows that it can be applied for practical use.

The second contribution in the area of scanned document images is segmentation of stamps in administrative documents. A stamp also serves as a seal for documents authenticity. However, the location of stamp on the document can be more arbitrary than a signature depending on the person sealing the document. This thesis shows that a

system based on our generic framework is able to extract stamps of any arbitrary shape and color. The evaluation of the presented system on a publicly available dataset shows that it is also able to segment black stamps (that were not addressed in the past) with a recall and precision of 83% and 73%, respectively.

The third contribution in the scanned document images is in the domain of information segmentation in technical drawings (architectural floorplans, maps, circuit diagrams, etc.) containing usually a large amount of graphics and comparatively less textual components. Further, as in technical drawings, text is overlapping with graphics. Thus, automatic analysis of technical drawings uses text/graphics segmentation as a pre-processing step. This thesis presents a method based on our generic information segmentation framework that is able to detect the text, which is touching graphical components in architectural floorplans and maps. Evaluation of the method on a publicly available dataset of architectural floorplans shows that it is able to extract almost all touching text components with precision and recall of 71% and 95%, respectively. This means that almost all of the touching text components are successfully extracted.

In the area of hyper-spectral document images, two contributions have been realized. Unlike normal three channels RGB images, hyper-spectral images usually have multiple channels that range from ultraviolet to infrared regions including the visible region. First, this thesis presents a novel automatic method for signature segmentation from hyper-spectral document images (240 spectral bands between $400 - 900$ nm). The presented method is based on a part-based key point detection technique, which does not use any structural information, but relies only on the spectral response of the document regardless of ink color and intensity. The presented method is capable of segmenting (overlapping and non-overlapping) signatures from varying backgrounds like, printed text, tables, stamps, logos, etc. Importantly, the presented method can extract signature pixels and not just the bounding boxes. This is substantial when signatures are overlapping with text and/or other objects in image. Second, this thesis presents a new dataset comprising of 300 documents scanned using a high-resolution hyper-spectral scanner. Evaluation of the presented signature segmentation method on this hyper-spectral dataset shows that it is able to extract signature pixels with the precision and recall of 100% and 79%, respectively.

Further contributions have been made in the area of camera captured document images. A major problem in the development of Optical Character Recognition (OCR) systems for camera captured document images is the lack of labeled camera captured document

images datasets. In the first place, this thesis presents a novel, generic, method for automatic ground truth generation/labeling of document images. The presented method builds large-scale (i.e., millions of images) datasets of labeled camera captured / scanned documents without any human intervention. The method is generic and can be used for automatic ground truth generation of (scanned and/or camera captured) documents in any language, e.g., English, Russian, Arabic, Urdu. The evaluation of the presented method, on two different datasets in English and Russian, shows that 99.98% of the images are correctly labeled in every case.

Another important contribution in the area of camera captured document images is the compilation of a large dataset comprising 1 million word images (10 million character images), captured in a real camera-based acquisition environment, along with the word and character level ground truth. The dataset can be used for training as well as testing of character recognition systems for camera-captured documents. Various benchmark tests are performed to analyze the behavior of different open source OCR systems on camera captured document images. Evaluation results show that the existing OCRs, which already get very high accuracies on scanned documents, fail on camera captured document images. Using the presented camera-captured dataset, a novel character recognition system is developed which is based on a variant of recurrent neural networks, i.e., Long Short Term Memory (LSTM) that outperforms all of the existing OCR engines on camera captured document images with an accuracy of more than 95%.

Finally, this thesis provides details on various tasks that have been performed in the area closely related to information segmentation. This includes automatic analysis and sketch based retrieval of architectural floor plan images, a novel scheme for online signature verification, and a part-based approach for signature verification. With these contributions, it has been shown that part-based methods can be successfully applied to document image analysis.

Contents

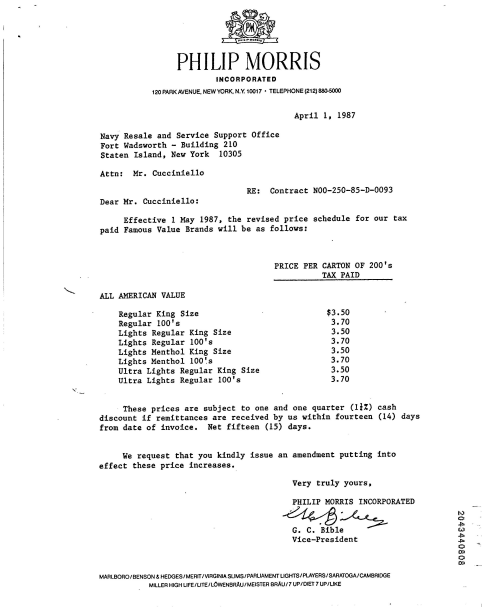
List of Figures

List of Tables

1 Introduction

Documents are usually considered as an integral part of various formal and informal workflows. For instance, an insurance invoice is an important document to claim expenditures from insurance companies; bank checks and deposit slips are used to withdraw and deposit money to/from banks; and personal notes are used to list down different structured and unstructured information like, recipes, lecture notes, shopping lists, etc. Traditionally, paper documents have remained in common use, however, with the evolution of modern computing technologies, paper documents are now being replaced by digitized (digital/scanned version of a traditional paper document) and born-digital documents.

In the contemporary work environment, automatic document processing systems are rapidly becoming necessary to process a large number of documents (both digitized and born-digital) efficiently. The evolution of technology and research in document analysis has enabled the development of various automatic systems capable of analyzing, processing, and understanding the content of given document images. A very common example of automatic document processing is that of an Optical Character Recognition (OCR) system that takes a document image, recognizes the printed/handwritten text available in the document, and saves it as a text file. This text can then be passed to other automatic data processing systems for further processing. Typically, a document image contains different types of information, for instance, text (machine printed/handwritten), graphics, signatures, and stamps. Figure 1.1 shows some example documents containing different types of information. The content needed by automatic document and data processing systems vary based on the type of processing they are designed for. Some need the complete content, while others need only a part of content extracted from a document image. For instance, a document archiving system [6] needs the complete content



PHILIP MORRIS
INCORPORATED
120 PARK AVENUE, NEW YORK, N.Y. 10017 • TELEPHONE (212) 880-5000

April 1, 1987

Navy Resale and Service Support Office
Fort Wadsworth - Building 210
Staten Island, New York 10305

Attn: Mr. Cucciniello

RE: Contract N00-250-85-D-0093

Dear Mr. Cucciniello:

Effective 1 May 1987, the revised price schedule for our tax paid Famous Value Brands will be as follows:

	PRICE PER CARTON OF 200's TAX PAID
ALL AMERICAN VALUE	
Regular King Size	$3.50
Regular 100's	3.70
Lights Regular King Size	3.50
Lights Regular 100's	3.70
Lights Menthol King Size	3.50
Lights Menthol 100's	3.70
Ultra Lights Regular King Size	3.50
Ultra Lights Regular 100's	3.70

These prices are subject to one and one quarter (1¼) cash discount if remittances are received by us within fourteen (14) days from date of invoice. Net fifteen (15) days.

We request that you kindly issue an amendment putting into effect these price increases.

Very truly yours,

PHILIP MORRIS INCORPORATED

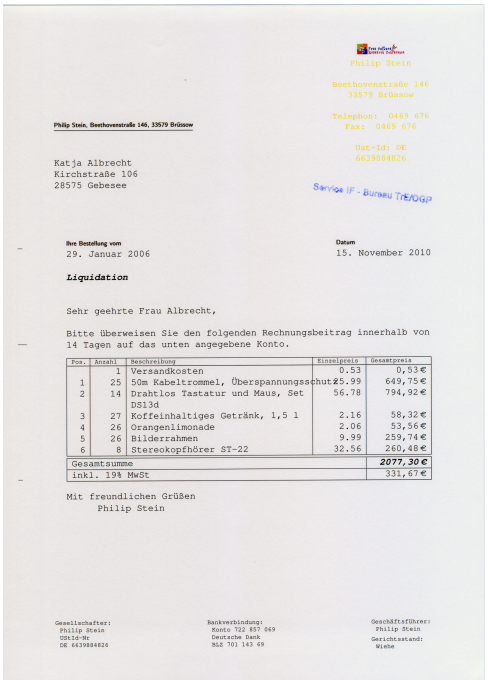
G. C. Bible
Vice-President

MARLBORO / BENSON & HEDGES / MERIT / VIRGINIA SLIMS / PARLIAMENT LIGHTS / PLAYERS / SARATOGA / CAMBRIDGE
MILLER HIGH LIFE / LITE / LÖWENBRÄU / MEISTER BRÄU / 7 UP / DIET 7 UP / LIKE

2043440808

(a) Letter with text, logo, and signatures

(b) Architectural floorplan with text and graphics



Philip Stein
Beethovenstraße 146
33579 Brüssow
Telephon: 0469 676
Fax: 0469 676
Ust-Id: DE
6639884826

Philip Stein, Beethovenstraße 146, 33579 Brüssow

Katja Albrecht
Kirchstraße 106
28575 Gebesee

Ihre Bestellung vom
29. Januar 2006

Datum
15. November 2010

Liquidation

Sehr geehrte Frau Albrecht,

Bitte überweisen Sie den folgenden Rechnungsbeitrag innerhalb von 14 Tagen auf das unten angegebene Konto.

Pos.	Anzahl	Beschreibung	Einzelpreis	Gesamtpreis
	1	Versandkosten	0.53	0,53€
1	25	50m Kabeltrommel, Überspannungsschutz	25.99	649,75€
2	14	Drahtlos Tastatur und Maus, Set DS13d	56.78	794,92€
3	27	Koffeinhaltiges Getränk, 1,5 l	2.16	58,32€
4	26	Orangenlimonade	2.06	53,56€
5	26	Bilderrahmen	9.99	259,74€
6	8	Stereokopfhörer ST-22	32.56	260,48€
Gesamtsumme				*2077,30€*
inkl. 19% MwSt				331,67€

Mit freundlichen Grüßen
Philip Stein

Gesellschafter: Philip Stein, USt1d-Nr DE 6639884826
Bankverbindung: Konto 722 857 849, Deutsche Bank, BLZ 701 143 69
Geschäftsführer: Philip Stein, Gerichtsstand: Wiehe

(c) Invoice with text, logo, and stamp

(d) Map with text

Figure 1.1: Sample document images with different types of information

of a document. On the other hand, an account management system may only need total amount, sender, receiver, and order details extracted from an invoice image. Similarly, a document management system [7] may only use a sender and receiver information to track and store documents. If these systems are presented with the complete content of a document image (containing some of the required and non-required contents), this might lead to errors. For example, if the recognized text of a bank receipt is given to an account management system, the system would face difficulty in first extracting the required information (account number, amount, purpose, signatures, etc.) from the document and may show an unexpected behavior. Hence, the integration of documents in automatic processing systems requires pre-segmentation of different types of information available in these documents.

This thesis presents a novel, generic framework for information segmentation from document images. The generic nature of the presented framework makes it applicable to a variety of documents (technical drawings, magazines, administrative, scientific, and academic) digitized using different methods (scanners, RGB cameras, and hyper-spectral imaging devices). The said framework is based on part-based features (SURF [8], BRISK [2], ORB [9], and FREAK [3]) and is capable of segmenting information located anywhere in a document image.

1.1 Motivation

The main motivation for this work comes from the author's observation that despite of their immense need in research and industry, there is a very limited availability of robust automatic information segmentation systems. Accordingly, the main objective achieved through this thesis is a generic framework, capable of segmenting different types of information (text, graphics, stamps, signatures) present in document images. This generic framework is based on the observation that different types of information available in documents have some very specific and unique properties (e.g., texture, intensity gradient), which could be utilized to differentiate them from one another. Humans are very good at differentiating objects based on these properties [10]. The aim is to develop automatic methods, which could identify and learn these properties and use them to segment different types of information available in document images.

The generic information segmentation framework presented in this thesis utilizes the so-

called part-based features, e.g., Speeded Up Robust Features (SURF) [8], and Fast Retina Keypoint (FREAK) [3]. The reason for considering these features is their ability to focus different parts of documents exclusively. This enables segmenting information located anywhere on a document image. Note that, the presented framework is not limited to any particular part-based feature; rather any available part-based feature approach can be used depending upon the need of efficiency and performance.

Another motivation of the generic framework is to enable information segmentation not only in scanned document images, but also in hyper-spectral documents. Forensic examiners use hyper-spectral imaging for analysis of documents while establishing document authenticity. However, in many cases, they have to manually segment signatures/stamps from documents. The presented framework is generic and is able to segment information from scanned as well as hyper-spectral document images.

Furthermore, the author noted that character recognition in camera-captured documents is a very challenging problem since these documents contain different camera specific distortions (e.g., blur, perspective distortion, and occlusion) which usually do not appear in the scanned images. Various Optical Character Recognition (OCR) systems are already available, but their behavior on camera-captured documents is still unknown. This is due to the lack of availability of any dataset containing camera captured document images. Manual labeling of camera-captured document is very laborious and time-consuming activity. However, a large dataset is required to train character recognition systems and gauge their performance. Therefore, a method is needed, which can automatically generate ground truth for words and characters in camera captured document images. This motivated the author to introduce a generic (independent of a particular language) and novel method for automatic ground truth generation of camera captured document images.

1.2 Problem Statement

This thesis addresses the problem of information segmentation in document images. It is very challenging as the type, location, properties, shape, and size of information may vary a lot in different documents, e.g., in some documents the information of interest may exist in the header while in others it may exist at some arbitrary location.

Information segmentation is required to integrate a document image into different auto-

matic data processing systems. In addition to textual information, automatic systems need non-textual information available in the document, e.g., stamps and signatures (considered as a seal of authenticity for documents). Almost all official documents, e.g., financial, governmental, security documents, bank checks, and even utility bills, are sealed with stamps and/or signatures. It is required to ensure the authenticity of documents (to check whether they are genuine or not) before integrating them into the normal work flow of any process, e.g., before issuing cash on bank checks, before releasing the claimed amount on invoices received by insurance companies. The signature and stamp verification systems are usually designed and trained to work on pre-segmented signature/stamp image. This means that they do not expect a complete document, but only signature/stamp image. However, in the real world, signature/stamp is a part of a document. In addition, their location on a document (especially in administrative documents) can be arbitrary and changes depending upon the content and the person who is putting a seal on a document. Therefore, information segmentation plays a vital role to enable use of document image information in different automatic data processing systems.

Furthermore, it is not possible to train a system with all of the possible occurrences of the target type of information. For instance, to enable signature segmentation in document images, it is not possible to train a system, which covers all possible signatures from everyone. Similarly, stamps can be of different types, shapes, and sizes, and it is not possible to train a system with every possible stamp which can appear. In addition to the different types of information, a document can be digitized via different digitization methods, e.g., with normal flatbed scanners, captured via RGB camera, and captured via hyper-spectral (in forensic environments) camera. Each of the digitization method has its inherent distortions, which makes it difficult to analyze and segment the information. Therefore, there is a strong need of a robust method, which can quickly learn and segment different types of information in a document digitized using different digitization methods.

In addition to information segmentation, this thesis also addresses the problem of generating automatic ground truth for large-scale camera captured document images. A major problem in the development of Optical Character Recognition (OCR) systems for camera captured document images is the lack of labeled camera captured document images datasets. The manual labeling of each word and/or character in captured images seem impractical for being very laborious and costly. Hence, there is a strong need of automatic methods capable of generating datasets from real camera-captured document images.

1.3 Hypotheses

The problems addressed in this thesis and the envisaged ideas lead to the formulation of the following hypotheses.

H1. It is possible to develop a generic part-based approach which is capable of segmenting different types of information in document images.

H2. The part-based approach can be adapted to fulfill task-specific computational and performance requirements.

H3. The part-based approach can be leveraged to cope with hyper-spectral document images.

H4. The use of part-based approach facilitates the automatic generation of large-scale ground-truth for related document image analysis tasks.

H5. A large dataset enables to perform deep learning even when the ground-truth has been generated automatically and contains errors.

1.4 Contributions

The main contribution of this thesis is the conceptualization and implementation of an information segmentation framework that is based on part-based features. The generic nature of the presented framework makes it applicable to a variety of documents (technical drawings, magazines, administrative, scientific, and academic documents) digitized using different methods (scanners, RGB cameras, and hyper-spectral imaging (HSI) devices). A highlight of the presented framework is that it does not require large training sets, rather a few training samples (for instance, four pages) lead to high performance, i.e., better than previously existing methods.

This thesis is divided into three major parts based on document digitization method (scanned, hyper-spectral imaging and camera captured) used. The thesis contributions in each of these parts are as follows:

1.4.1 Scanned Document Images

In the area of scanned document images, three specific contributions have been realized. These contributions validate Hypotheses H1 and H2.

1. This thesis presents a novel method for signature segmentation in administrative documents. In some workflows, it is very important to check a document's authenticity before processing the actual content. This can be done based on the available seal of authenticity, e.g., signatures. However, general signature verification systems expect a pre-segmented signature image. To use signature verification systems on document images, it is necessary to first segment signatures in documents. This thesis shows that the presented framework can be used to segment signatures in administrative documents. The system based on the presented framework is tested on a publicly available dataset where it outperforms the state-of-the-art methods and successfully segments all signatures, while less than half of the found signatures are false positives. This shows that it can be applied for practical use.

2. This thesis presents a novel method for stamp segmentation in administrative documents. A stamp also serves as a seal for documents authenticity. However, the location of stamp on a document can be more arbitrary than a signature depending on the content of a document and person sealing that document. This thesis shows that a system based on the presented generic framework is able to extract stamps of any arbitrary shape and color. The evaluation of the presented system on a publicly available dataset shows that it is also able to segment black stamps (that were largely neglected in literature in the past) with a recall and precision of 83% and 73%, respectively.

3. This thesis also contributes in the domain of information segmentation in technical drawings, e.g., architectural floorplans, maps, and circuit diagrams. Such documents usually contain large amounts of graphical and comparatively less textual components and text is generally overlaps with graphics. Thus, automatic analysis of such documents requires text/graphics segmentation as a pre-processing step. This thesis contributes a method based on the presented generic information segmentation framework that is able to detect text, which is touching graphical components in architectural floorplans and maps. Evaluation of the method on a publicly available dataset shows that it is able to extract almost all touching text components with precision and recall of 71% and 95%, respectively. This means

that almost all of the touching text components are successfully extracted.

1.4.2 Hyper-spectral Document Images

Two contributions have been made in the area of hyper-spectral document images, which validate hypothesis H3.

1. This thesis presents a novel automatic method for signature segmentation from hyper-spectral document images (240 spectral bands between 400 - 900 nm). The proposed method is based on a part-based key point detection technique, which does not use any structural information, but relies only on the spectral response of the document regardless of ink color and intensity. The presented method is capable of segmenting (overlapping and non-overlapping) signatures from varying backgrounds like, printed text, tables, stamps, etc. Importantly, the presented method can extract signature pixels and not just the bounding boxes. This is substantial when signatures are overlapping with text and/or other objects in an image.

2. This thesis presents a new dataset comprising of 300 documents scanned using a high-resolution hyper-spectral scanner. Evaluation of the presented method on this hyper-spectral dataset shows that it is able to extract signature pixels with the precision and recall of 100% and 79%, respectively.

1.4.3 Camera Captured Document Images

Two main contributions have been made in the area of camera captured document images. These contributions validate Hypotheses H4 and H5.

1. This thesis presents a novel, generic, method for automatic ground truth generation of document images. The presented method builds large-scale (i.e., millions of images) datasets of labeled camera captured / scanned documents without any human intervention. The method is generic and can be used for automatic ground truth generation of (camera captured and/or scanned) documents in any language, e.g., English, Russian, Arabic, Urdu. The evaluation of the presented method, on two different datasets in English and Russian, shows that 99.98% of the images are correctly labeled in every case.

2. This thesis presents a large dataset comprising 1 million word images (10 million character images), captured in a real camera-based acquisition environment, along with the word and character level ground truth. This dataset can be used for training as well as testing of character recognition systems for camera-captured documents. Various benchmark tests are performed to analyze the behavior of different open source OCR systems on camera captured document images. Evaluation results show that the existing OCRs, which already get very high accuracies on scanned documents, fail on camera captured document images. Using the presented camera-captured dataset, a novel character recognition system is developed which is based on a variant of recurrent neural networks, i.e., Long Short Term Memory (LSTM) that outperforms all of the existing OCR engines on camera captured document images with an accuracy of more than 95%.

1.4.4 Related Topics

Several contributions have been made in the related application areas.

1. A system for automatic analysis and sketch based retrieval of architectural floor plans is presented. This system is a perfect example of systems, which needs segmented information at different points of time during analysis.
2. A novel framework for real-time online signature verification. This presented framework has been applied on different scenarios, including signatures in financial contracts or ordering processes.
3. This thesis also presents a novel part-based approach for offline signature verification system.

1.5 Thesis Structure

The rest of the thesis is structured as follows. Chapter 2 provides detailed insights into important aspects and the state-of-the-art of information segmentation in document images. In this context, an overview of different methods available for signature, stamp, and text/graphics segmentation in document images is given. Chapter 3 provides details on part-based features which are used through the course of this thesis. These local features form the foundation of the generic information segmentation framework presented in this

thesis. Chapter 4 provides theoretical and algorithmic aspects of this generic information segmentation framework.

The rest of this thesis is organized into four main parts based on the document digitization method. Those are scanned documents, camera captured documents, and documents captured through hyper-spectral imaging. The fourth part discusses contributions in related topics. Details of each part follow in the remainder of this section.

The first part of this thesis is dedicated to the analysis of scanned document images. In particular, this part is focused on information segmentation in scanned administrative documents and technical drawings. Chapter 5 provides details of the method presented in this thesis for signature segmentation in administrative documents. The presented signature segmentation method is based on the information segmentation framework presented in Chapter 4. The method outperforms state-of-the-art methods for signature segmentation in document images. Chapter 6 presents a method for segmentation of stamps in administrative document images. The presented method is also based on the framework presented in Chapter 4 and able to segment different seen and/or unseen stamps in any color, shape, and size, located anywhere in a document. Chapter 7 presents a novel method for text/graphics segmentation in document images. The documents focused in this chapter are technical drawings, e.g., architectural floor plans and maps.

The second part of the thesis is dedicated to information segmentation in specialized documents, i.e., hyper-spectral document images, commonly used by forensic document examiners (FDEs). Chapter 8 presents a method for signature segmentation in hyper-spectral document images. In addition, to a segmentation method, this chapter also presents a novel dataset consisting up of 300 hyper-spectral documents containing text, signatures, stamps and other information.

The third part of the thesis is dedicated to camera-captured document image analysis. The contribution of Chapter 9 is many-fold. The first contribution is a novel, generic method for automatic ground truth generation of camera-captured document images (books, magazines, articles, invoices, etc.). It enables building large-scale (i.e., millions of images) labeled camera-captured/scanned document dataset without any human intervention. The second contribution is a large dataset (called C^3Wi) of camera-captured characters and words images, comprising of one million word images (ten million character images), captured with cameras. The third contribution is a novel method for the recognition of camera-captured document images. The presented method is based

on Long Short-Term Memory and outperforms the state-of-the-art methods for camera based OCRs. As a final contribution of this chapter, various benchmark tests are performed to uncover the behavior of commercial (ABBYY), open source (Tesseract), and the presented camera-based OCR using the presented C^3Wi dataset.

The last part of the thesis is dedicated to other related topics which are explored during the span of this thesis. It includes automatic analysis and sketch based retrieval of architectural floor plans (Chapter 10), a novel framework for online signature verification system (Chapter 11), and a part-based system for offline signature verification (Chapter 12). In the end, Chapter 13 summarizes the main conclusions of this thesis and provides an outlook on future research.

2

Information Segmentation in Document Images: The State-of-the-Art

A document image contains different types of information, for instance, text (machine printed/handwritten), graphics, signatures, and stamps. Different automatic document image analysis (DIA) systems are available and designed for specific tasks e.g., OCRs, signature verification system, and invoice processing system. It is important to segment different types of information available on a document so that only required information is passed to subsequent analysis processes rather than the complete content of document. This chapter provides detailed insights into the various important aspects and the state-of-the-art of information segmentation in document image. In this context, one of the most important types of information, which appears in most of the legal and administrative documents, is signature, which is considered as a seal of authenticity. Section 2.1 provides an overview of the state-of-the art methods available for signature segmentation in document images. Stamps (another important seal of authenticity) appear on many administrative and legal documents. In order to analyze these stamps for either verification or for document retrieval, it is important to first segment them from documents. Section 2.2 provides an overview of different stamp segmentation methods available in the literature. Finally, Section 2.3 is dedicated to the state-of-the-art in text/graphics segmentation, which is very common in technical drawings, e.g., architectural floorplans, and circuit diagrams.

Figure 2.1: Stamps in different colors

2.1 Stamp Segmentation

Various methods have been presented for segmentation of stamps from document images. Some use color and geometric features to separate stamps from the other content present in document images, whereas others view the task as an object recognition problem.

Ueda [11] presented an approach for detection and extraction of signatures and seals using color information from bank checks. It is assumed that color of signatures, seals, and background pattern is different; therefore, three different clusters are extracted. A major problem with this approach is the assumption that only three different clusters exist on checks, which is not always the case. A more general approach working on more clusters has been introduced by Soria-Frisch [12]. He used fuzzy integral for the selective extraction of color clusters. However, also in this work it is assumed that stamps are of a single color, which in itself is not always true (see Figure 2.1).

Cai and Mei [13] presented a method to detect and verify seal/stamp imprints. However, it is required to register the seal/stamp in prior. This means that seal/stamp template must be available apriori, which is not possible in many cases.

A framework for segmenting stamps from document images using characteristics of stamp patterns is presented by Zhu et al. [14]. It is based on estimation of connected edge features. Although it is able to segment even overlapping stamps, but it is limited to elliptic/circular (oval) shaped stamps. Chen et al. [15] presented an approach for detection of seal imprints on Chinese bank checks. It is based on the region growing approach, where regions are first located using region growing algorithm and later fused together. The method is only able to deal with the seals if no other objects (e.g., logos) are present on checks.

Roy et al. [16] presented a method for seal/stamp object detection from documents with cluttered background. They view the problem as object recognition and used General-

ized Hough Transform [17] along with voting to find the possible location of seal object based on spatial feature description. This approach focuses on elliptical, circular, and rectangular stamps and need to have a template of stamp beforehand similarly to Cai and Mei [13].

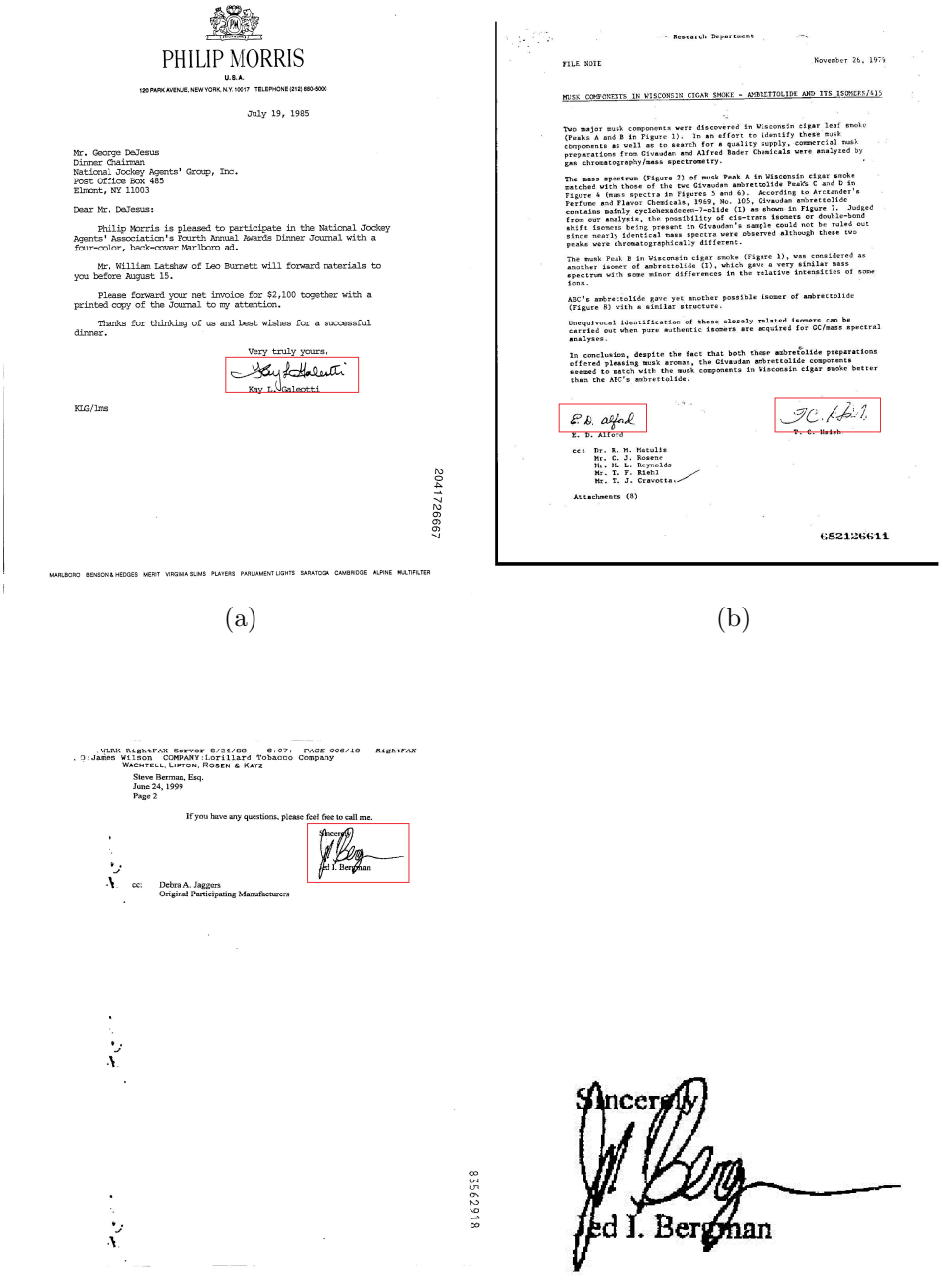
An approach for detection, localization, and segmentation of stamps in the scanned documents is presented by Frejlichowski and Forczmańnski [18]. This method is flexible in terms of detectable stamp shapes, and is not limited to only rectangular and oval shapes. It is based on the analysis of documents in YC_bC_r color space using horizontal and vertical projection profiles and five different shape descriptors. As documents are processed in YC_bC_r color space, the method is limited only to color stamps. In addition, it is unable to detect the textual stamps which do not contain any regular or irregular shape around them.

Recently, an automatic method to segment and verify stamps from document images was presented by Micenkova and van Beusekom [19]. It is based on the segmentation of image by color clustering in YC_bC_r color space and classification of candidate solutions by geometrical and color-related features. One of the main advantages of this approach is that it is not restricted to stamps of a particular shape. One of the main advantages of this approach is that it is not restricted to stamps of a particular shape. In addition, it is also capable of detecting textual stamps. Similar to Frejlichowski and Forczmańnski [18], they have also performed a complete analysis in YC_bC_r and are not able to deal with black stamps.

From the above short survey, it is evident that in the past, mostly color stamps remained under focus. Only the method presented by Zhu et al. [14] can detect black stamps as well, however, it is limited to oval shaped stamps only.

2.2 Signature Segmentation

Up to the best of author's knowledge, currently there exists no survey for signature segmentation. A survey on banks check processing is the most relevant one presented by Jayadeven et al. [20]. However, this survey does not cover the area of signature segmentation methods. Jayadevan et al. [21] presented a method for analysis of signatures present on bank checks. The signatures are extracted from a check image by dividing the image into four equal quadrants and selecting the lower right quadrant. This selection is



(a) (b)

(c) (d)

Figure 2.2: (a), (b), (c) Signatures at different positions in document images, (d) Signature overlapping with text

based on the assumption that signatures will always appear only in the lower right quadrant and this quadrant does not contain anything else except signatures and background. Furthermore, it is assumed that signatures will always be making a positive angle with the horizontal axis. These assumptions are valid only in few cases and only for specific check templates. Djeziri et al. [22] and Madasu et al. [23] specifically presented methods for extraction/segmentation of signatures. Djeziri et al. [22] presented a method which is inspired from human visual perception and based on filiformity criteria. Contour lines of objects are differentiated from handwritten lines using the filiformity criteria. Madasu et al. [23] used a sliding window to calculate the entropy and to fit the window to signature block. Sankari et al. [24] presented an approach for segmentation of bank check account number and account holders signature from check images using prior knowledge of Cartesian coordinate space. These segmented regions are further used for training and verification using Hamming distance measures.

In case of bank checks, a priori information about the location of signatures is already available, which makes the segmentation process comparatively easy. Therefore, most of the existing system for signature verification can be optimized and applied directly to bank checks if this prior information is available. In case where the prior information regarding particular location/position of signatures in a document is not known, application of current automatic signature verification systems is still a challenge.

There are plenty of documents other than bank checks, which also contain signatures, e.g., contracts, invoices, payslips, and wills. Figure 2.2 shows that segmentation in these document images becomes more challenging as signatures can be at different parts of documents depending upon the content. To deal with signatures in complete documents, Zhu et al. [25] presented a method for segmenting signatures from complete documents using saliency map and a publicly available dataset namely Tobacco-800. Details of the Tobacco-800 dataset can be found in Section 5.3. A complete document retrieval system is presented by Zhu et al. [26] where signature matching is combined with detection framework from [25]. Mandal et al. [27] presented an approach for signature segmentation using conditional random fields. Results are reported on a subset of Tobacco-800 dataset, i.e., 105 images out of 1290. One of the main problems of this approach is that it requires a large number of training samples. Mandal et al. [27] segment signatures on patch level, whereas in some cases, some text is touching the signature components which may cause problems later in signature verification.

To rectify the problem of touching characters, Partha et al. [28] presented an approach

for segmentation of signatures along with the segmentation of those characters which are touching signature strokes. They used gradient-based features (grayscale local orientation histogram) with SVM as classifier to classify blocks as signature or printed text. To remove touching characters from signature stroke, hypothetical zones are detected, where possible overlapping characters may exist, using neighboring printed blocks. Contours smoothness information near skeleton junction is used to separate text from characters.

A major problem with all of the above-mentioned approaches of signature segmentation, from document images, is that none of them is applied on the complete dataset (Tobacco-800). Subsets of Tobacco-800 dataset are used and it is not mentioned that which images are included in any particular subset. This makes it difficult to find the behavior of these methods in case of some other existing classes. In addition, none of the above-mentioned systems reported efficiency of system in term of time and complexity.

Furthermore, some commercial systems capable of finding one or two signatures on bank checks as well as in IRD[1] images and snippets, SignatureXpert-2[2] by Parascript. The problem with commercial systems is that the data on which they are trained is never made publicly available . Also details about how these systems are working and the methods they are applying are not known.

2.3 Text/Graphics Segmentation

Several different methods have been presented to perform text/graphics segmentation for different scenarios. Initially, the focus was to retrieve only the text in general document images which is not touching graphics. Wahl et al. [29] presented a method for block segmentation and text extraction in mixed text/image documents. This method has shown promising results for text line segmentation and image block separation. Various improvements have been presented for this approach. Bukhari et al. [30] used a self-tunable Multilayer Perceptron (MLP) classifier for distinguishing between text and non-text connected components using shape and context information as a feature vector. The presented system is evaluated on a subset of the UW-III, the ICDAR 2009 page segmentation competition [31] test images, and circuit diagrams dataset. In another work, Bukhari et al. [32] adapted the Bloomberg's text/halftone segmentation algorithm

[1] An Image Replacement Document (IRD) is a replacement check on paper, generated from the electronic image of an authentic chapter check.

[2] http://www.parascript.com/recognition-products/forms-processing/signaturexpert-2

to make it applicable to text and non-text image segmentation approach where drawings, maps, and graphs are considered as a halftone.

Garg et al. [33] presented a color based clustering and Conditional Random Fields (CRF) based for text/graphics segmentation. The classification of pixels is done by a top-level CRF based on the semantic correlation learned across clusters. However, the presented method works only with color document images. Priti P Rege [34] presented a method where first a document is segmented into several non-overlapping blocks and then the features (connectivity histogram and image boundary/perimeter) are extracted for each of the segmented blocks. This method is a combination of run-length smearing and boundary perimeter detection. Finally, each block is classified as text/non-text component.

Garg et al. [35] presented a method for text/graphics segmentation in Indian newspapers using Expectation Maximization and edge direction histogram features. Li et al. [36] presented a two-stage method for segmentation of text in document images. First, the texture features are extracted for each block based on Gabor filter. Second, classification of the texture feature is performed using kernel based self-optimization Fisher classifier.

Vu et al. [37] presented a method for text extraction from document images with the aim to use it for image database indexing, document understanding, and image-based web searching. They presented a three-stage approach to extract text from document images. This approach identifies text which is relatively large in size and has high contrast. A document is divided into small blocks and K-mean clustering is used on these blocks after multilevel thresholding. Finally, a connected-component based filtering is used to separate text from all other objects in a document.

Maji and Roy [38] presented a text/graphics segmentation method based on M-band wavelet packet analysis (to extract the scale-space features) and rough-fuzzy-possibilistic c-means (to address the uncertainty problem). In addition, they use unsupervised feature selection to select relevant and non-redundant features. The presented method is invariant to font size, line orientation, and script of the text.

The focus of all of the above-mentioned approaches is on "general" DIA. However, text-graphics segmentation plays a very important role in technical drawings, e.g., floorplan, circuit diagram, and chemical structure diagrams. Chapter 10 presents a complete system for analysis of architectural floorplans, where text/graphics segmentation is an important part of the complete analysis. Specifically, for technical drawings, Fletcher and Kasturi [39] presented a method for separation of text strings from mixed text/graphics

images which is based on connected component analysis. A major advantage of this approach is its simplicity, as it is based on filtering of connected components based on their aspect ratio. It gives promising results on the images where no text is touching the graphics. However, in most of the technical drawings and map images, text and graphics overlay. A minor drawback of their method is that the text touching graphics are marked as graphics component rather than text.

Dori and Wenyin [40] performed vector-based segmentation to extract text connected to graphical elements. The focus of this work is engineering drawings. The method is based on growing individual character box regions, which are then merged into text boxes. Finally, the text boxes are re-segmented into character boxes.

A text/graphics separation method for overlapping text and graphics was presented by Cao and Tan [41]. They start with pre-processing to separate the solid graphical components and remove all the dashed lines. This method is also based on connected component analysis where a size filter is used for marking components as either text or graphics. They applied this method to images of maps.

Adam et al. [42] used Mellin Fourier Transform to classify characters and symbols drawn on technical drawings. A Filtering technique is used to detect touching characters and symbols. Most of the touching characters are successfully extracted by this method. A major disadvantage of this method is that it is very time consuming.

Tombre et al. [43] presented an extension of the method presented by Fletcher and Kasturi [39]. In addition to aspect ratio filter, they introduced additional size and shape filters, for connected components. Furthermore, they split image into three layers, i.e., text, graphic, small-elongated components layer. The third layer is used for finding the dashed lines and text string extraction. After text separation, Hough transform is used to group characters into strings. This method improved the results of [39], but still some touching characters were marked as graphics components.

Roy et al. [44] further extended the approach of Tombre et al. [43] by using color information to separate touching text from the graphics. After separation of text/graphics, Hough transform is used to remove the lines from the image. Finally, pyramid segmentation is used for grouping the characters into words. This method can be used where text and graphics are occurring in different colors.

Jafri et al. [45] introduced a hierarchical method for segmenting text areas in natural images. The basic assumption is that text is written with a contrast color on a uniform

background. Segmentation process starts with finding the text background areas. In each segmented region, the presence of text is tested afterwards.

Raveaux et al. [46] used color information coupled the graph representation. Initially, a color model is computed from the color properties of the image. Then image contours are extracted using edge detection. Finally, connected components of the contour image are classified according to the graph representation. Structural training is used to learn the text and graphics diversity. They also based their method on the assumption that text is not touching graphics components.

Roy et al. [47] used the SIFT features for extraction of text touching graphics. The SVM classifier is used to extract non-touching text. To extract the touching characters SIFT features of template and image are compared. To accommodate rotation of characters, shape models are used. The presented method is tested on geographical map images and is capable of extracting most of the touching characters.

Hoang and Tabbone [48] introduced an approach for text/graphics segmentation. This technique is based on the sparse representation framework and two appropriately chosen discriminative dictionaries. Using each dictionary, sparse representation of one type of signal and non-sparse representation for other type of signal is achieved. Finally, text, graphics separation is achieved by promoting the sparse representation of input image in these two dictionaries. Hoang and Tabbone [48] claimed that their approach could be used in any domain. However, an adaption to floor plans would require additional effort in order to benefit from specific properties of floor plan images.

Ahmed et al. [49] presented a method for text/graphics segmentation in architectural floor plans. This method extends the method presented by Tombre et al. [43], by providing an automatic mechanism to calculate different thresholds. This method has good accuracy and is able to extract most of the overlapping text, but can only be used for architectural floor plan images.

Do et al. [50] presented a method to extract text areas from graphical documents using sparse representation and multi-learned dictionaries (for both text and graphics). For all the patches in dictionaries, a sparse representation is computed and is used this to classify a patch as text or graphics.

Mello and Machado [51] presented a method for text segmentation in vintage floor plans and topographic map images. The presented method is based on background removal, thresholding, histogram equalization, connected component analysis, line removal, noise

removal, and restoration.

3 Local Features for Information Segmentation

Local/part-based features have already shown good results for different problems, e.g., object recognition [52], object detection [53, 54], and object tracking [55]. In this thesis, local features are used for the problem of information segmentation in document images. In the past, different real valued as well as binary local/part-based approaches were presented. This chapter provides an overview of different local/part-based features used in this thesis. These features form the foundation of the information segmentation framework presented in this thesis. Section 3.1 provides an overview of real valued local/part-based features, i.e., Speeded Up Robust Features (SURF). Section 3.2 provides an insight of local/part-based keypoint detection technique, i.e., Features from Accelerated Segment Test (FAST). Section 3.3 to 3.6 are dedicated to different binary local/part-based features.

3.1 Speeded Up Robust Features (SURF)

Speeded Up Robust Features (SURF) is a one of the most commonly used part-based/local feature approach. This was primarily developed for object detection and recognition tasks in grayscale images. However, there are now many variants of SURF [56, 57], which are also applicable to color and hyperspectral images. SURF can be referred as an improved version of another very popular local keypoint detector and descriptor, i.e., Scale Invariant Feature Transform (SIFT) [58], however, SURF is several times faster, robust, and provide invariance to many image transformations than SIFT [8].

Localization of important areas (the so called "keypoints") in an image is a key for local/part-based features. An area in an image is referred as keypoint if it is significantly different from its neighbors. This difference can be computed based on change in gradients, edges, texture, etc. Once the locations of keypoints are known, the next step is to extract meaningful and representative features which best describe the information available in the area around the keypoint. This step is referred to as keypoint description.

SURF provides a robust method for both detection and description of keypoints in an image. The speeded nature of SURF is due to the use of a special image representation called integral images [59]. SURF uses integral images to detect blob like structures in an image/rectangular grid/patch. An integral image is a sum of the values between the point under consideration and the origin. Integral images are already used by many researchers to improve the efficiency of different algorithms, e.g., efficient adaptive thresholding [60], comparison of color images [61]. The use of integral images reduces the calculation of an area of rectangular region to only four operations, regardless of the size of area. Therefore, all of the convolutions with different filters, sums, can be done very efficiently, and this is the main fuel, which speeds up the SURF.

3.1.1 Keypoint Detection

SURF keypoint detector uses Hessian matrix (which is a matrix containing partial derivatives of a function) to detect keypoints in an image. Considering a 2D image I as a function f of two variable (x, y). The Hessian matrix is given as

$$H(I(x,y)) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \tag{3.1}$$

Based on the above mentioned equation, the determinant of the Hessian, which is also called as discriminant, is given by

$$det(H) = \frac{\partial^2 I}{\partial x^2}\frac{\partial^2 I}{\partial y^2} - \left(\frac{\partial^2 I}{\partial x \partial y}\right)^2 \tag{3.2}$$

In addition to the use of integral images, another speedup is achieved in SURF by using determinant of Hessian for both location and scale determination. The Hessian matrix

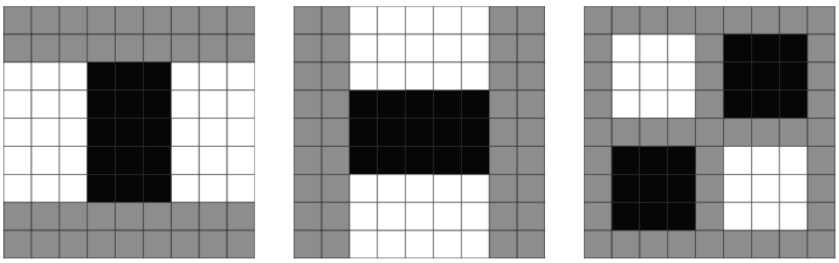


Figure 3.1: Weighted box filter approximations in x, y, and xy-directions

with scale and space can be rewritten as follows.

$$H(X,\sigma) = \begin{bmatrix} Dxx(X,\sigma) & Dxy(X,\sigma) \\ Dxy(X,\sigma) & Dyy(X,\sigma) \end{bmatrix} \tag{3.3}$$

In Equation 3.3 X is a point (x, y) in an image and σ is the scale. $Dxx(X, \sigma)$ is the partial derivative of image which is calculated using image convolution with scale normalized approximated Gaussian box filters. These box filters can be constructed for Gaussian derivatives in x, y and combined xy direction and can be computed very fast using the integral images. An approximated 9×9 weighted box filter in x, y and combined xy direction with the scale $\sigma = 1.2$ (which represents the highest spatial resolution and minimum scale) is shown in Figure 3.1. The determinant of the Hessian (Equation 3.2) is considered as a blob response at location (x, y, σ). A keypoint can be found in different scales, as it is possible that the same image appears in different resolutions. It is therefore important to have a keypoint detection mechanism, which is scale invariant. This is achieved by using scale space representation, where the keypoints are looked for both in scale and space. The scale-space is divided into a number of response maps layers, called octaves, where each layer covers the double of scale. The 9×9 box filter shown in Figure 3.1 corresponds to a real valued Gaussian filter of $\sigma = 1.2$. To obtain new layer the filter is up-scaled while maintaining the ratio. To filter out unnecessary keypoints, the responses on all of the detected layers are thresholded. This means that all those keypoints who have values less than some predefined threshold are removed. This threshold is also referred to as the Hessian threshold. On the remaining points, non-maximal suppression is preformed to find the set of probable candidate points. Finally, localization of keypoints in both scale and space is achieved by interpolation in the neighboring data by fitting 3D quadratic [62].

3.1.2 Descriptor

Once a keypoint is detected in an image, the next step is to perform measurements in that area and describe the content in the area using these measures. This step is also referred to as keypoint descriptor extraction phase. SURF uses location information and the distribution of gradient in the region of keypoint to describe the content in the region.

While describing the content of a keypoint, it is important to find the reproducible orientation assessment, so that the descriptor extracted for the given patch is rotation invariant. Orientation assignment is done using the Haar wavelet response in x and y directions in a circular neighborhood of radius $6s$, where s is the scale at which the keypoint was detected. The responses are again computed using integral images, which results into a speeded computation. These responses are then represented as a vector, and the dominant orientation is estimated by summing up all the responses within a sliding orientation window covering an angle of $\pi/3$. A variant of SURF, called Up-right SURF (U-SURF), does not perform this step, which results into a more speeded but non-rotation invariant version of SURF. Once the orientation of the keypoint is determined, the next step is to extract the descriptor itself, which is done as follows:

1. Construct a square window of size $20s$ centered on the keypoint, oriented along the estimated orientation.

2. Split the window in 4×4 sub regions.

3. Compute wavelet response in x and y direction for each sub region, referred to as dx and dy.

4. The sum of wavelet responses in each sub region is stored as a first component of the descriptor ($\sum dx$, $\sum dy$). To store the information about the polarity of the intensity change, sum of absolute values of response is also stored, i.e., $\sum \mathsf{dx}$, $\sum \mathsf{dy}$.

This results into a 4 dimensional descriptor for each sub region, which in-turn results into a 64 dimensional descriptor for a keypoint. Figure 3.2 shows description of a keypoint in an image.

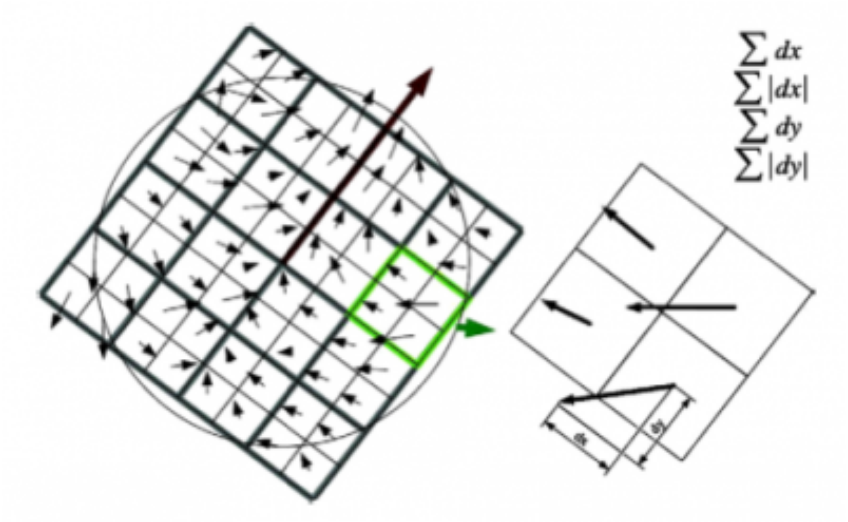


Figure 3.2: SURF keypoint description

3.2 Features from Accelerated Segment Test (FAST)

Keypoint detection is considered as an important step for many computer vision and image-processing tasks, e.g., object tracking, image matching, and object recognition. FAST [1] is an efficient method for identification of keypoints in an image. FAST keypoint detection uses machine learning to detect keypoints, which are stable, and computationally less expensive than many of the keypoint detection methods like, Difference of Gaussian [58], SUSAN [63], Harris [64], etc. It is based on segment test criteria, which works by analyzing 16 pixels around the keypoint candidate pixel p. These points are labeled clockwise by a number from 1 to 16 (see Figure 3.3).

A following check is performed on the candidate pixel, so that to decide whether to classify a point as a keypoint or not.

1. Set a threshold intensity value T.

2. If a set of N neighboring pixels in a circle have intensity greater than the candidate pixel intensity plus threshold intensity,
   OR If a set of N neighboring pixels in a circle has intensity less than the candidate pixel intensity minus threshold intensity, the candidate pixel is referred to as a keypoint.

The above process will return many keypoints adjacent to one another. To overcome this problem, non-maximal suppression is performed. However, to perform non-maximal

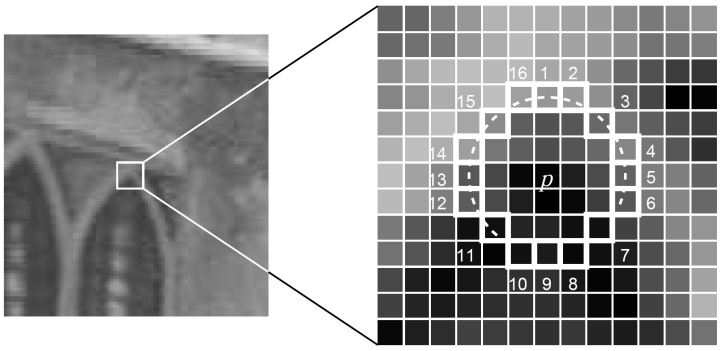


Figure 3.3: An example of segment test [1]

suppression, it is necessary to first compute a score for each keypoint. The score is computed (using equation 3.4) by the sum of absolute difference between the pixels in the neighboring arcs and the center pixel. Adjacent keypoints are compared based on this score and the one with lower score is removed.

$$V = max \begin{cases} \sum(pixelvalues - p) & (value - p) > t \\ \sum(p - pixelvalues) & (p - value) > t \end{cases} \tag{3.4}$$

3.3 Binary Robust Independent Elementary Features (BRIEF)

BRIEF is the first binary local/part based feature. The aim of developing BRIEF was to have a fast binary description of a patch around a keypoint. The motivation behind having a binary descriptor is the ease in computing and efficient matching using the Hamming distance measure. Similar to USURF, BRIEF is also not rotation invariant.

To generate binary descriptor, BRIEF performs a test (shown in equation 3.5) on patch around the keypoint [65]. To avoid effects of noise and have a better description, Gaussian smoothing (with a discrete kernel window of 9×9) is performed on the patch [65]. To generate binary descriptor, BRIEF performs a test (shown in equation 3.5) on patch around the keypoint [65]. To avoid the effects of noise and have a better description,

Gaussian smoothing (with a discrete kernel window of 9×9) is performed on the patch [65].

$$\tau(p;\ x, y) = \begin{cases} 1 & if\ I(p, x) < I(p, y) \\ 0 & otherwise \end{cases} \tag{3.5}$$

Where $I(p, x)$ is the intensity of pixel at $x = (u, v)^T$. To perform the above mentioned test it is important to first select (x, y)-location pairs uniquely [65] and based on these selected pairs a binary descriptor is defined by equation 3.6

$$f_{n_d}(p) = \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(p;\ x_i, y_i) \tag{3.6}$$

Where n_d refers to the number of dimensions of the binary descriptor, which can be $128, 256$ and, 512.

An important step to build a binary descriptor using equation 3.6 is to select the location pairs $(p;\ x_i, y_i)$. Note that BRIEF does not have any specific sampling pattern. To select the n pairs for building up a binary descriptor, authors of the paper experimented five different strategies, and selected the random sampling using a Gaussian distribution with $\mu = 0$ and $\sigma^2 = \frac{1}{25} S^2$ (locations closer to the center has higher preference) where S is the size of patch [65].

3.4 Oriented Fast and Rotated BRIEF Features (ORB)

ORB is a fusion of FAST [1] keypoint detector and a rotation invariant version of BRIEF [65]. Both FAST and BRIEF are popular due to their good performance and low computation cost. As mentioned in Section 3.3, BRIEF is not rotation invariant, ORB uses orientation estimation from FAST keypoints and efficiently computes oriented BRIEF [9].

3.4.1 Keypoint Detection with Oriented FAST

ORB uses FAST keypoint detector to detect initial keypoints using FAST-9 (i.e., with a circular radius of 9) [9]. After initial keypoint detection, Harris corner detection [64] is used to select top N keypoints out of all of the detected keypoints by sorting them based on Harris detection score. Furthermore, scale pyramid is built to produce Harris filtered FAST features at each level of the pyramid [9]. However, as mentioned in Section 3.2, FAST does not has any orientation assignment with it. To estimate the orientation, ORB uses intensity centroid of the patch. It assumes that a corner intensity is not at the center and uses the direction vector from center to corner to compute the orientation [9].

The centroid is computed using equation 3.8 defined by image moments (equation 3.7)

$$m_{pq} = \sum x^p y^q I(x, y) \tag{3.7}$$

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{3.8}$$

Finally, the orientation of the patch is defined by equation 3.9

$$\theta = atan2(m_{01}, m_{10}) \tag{3.9}$$

3.4.2 Rotated BRIEF Descriptor

The keypoint detection using oriented FAST result into keypoint along with orientation information. Originally BRIEF is not rotation invariant. In ORB, the rotation invariance is achieved by using a steer BRIEF according to the orientation of keypoint. This means that the patch is first oriented and then the binary pairs are selected. ORB constructs a lookup table of precomputed BRIEF patterns by discritizing the angle to increments of $2\pi/30$ (12 degrees). If the keypoint orientation θ is consistent across views, the correct set of points S_θ will be used to compute its descriptor [9].

Furthermore, ORB uses a learning step with aims to find less correlated binary pairs with high variance so that each pair refers to a new information which increases the amount of information encoded in the patch (whereas, high variance increases the discriminative power of the feature). To learn the best pairs, the authors of the ORB paper use 300,000

keypoints extracted from PASCAL 2006 dataset and use the greedy algorithm defined as follows:

- Run each test against all training patches.
- Order the tests by their distance from a mean of 0.5, forming the vector T.
- Greedy search:
    1. Put the first test into the resultant vector R and remove it from T.
    2. Take the next test from T, and compare it against all tests in R. If its absolute correlation is greater than a threshold, discard it; else, add it to R.
    3. Repeat the step 2 until there are 256 tests in R. If there are fewer than 256, raise the threshold and try again.

The resulting vector is referred as to rBRIEF, is significantly better than the original BRIEF and steered BRIEF.

3.5 Binary Robust Invariant Scalable Keypoints (BRISK)

BRISK is a local-part based method primarily developed for different computer vision applications. The motivation behind BRISK is to a provide high quality description with low computation cost [2]. It is much faster than SURF, in many cases; it is an order of magnitude faster than SURF, while having similar results as SURF. Similar to SURF, BRISK also provides a mechanism for keypoint detection as well as description. The speed gain in BRISK is due to the use of a variant of FAST [1] i.e., AGAST [66]. This means that in principal, the BRISK keypoint detection method is based on FAST. In addition, an important reason for the increase in speed is the binary nature of the descriptor. The descriptor is computed by simple comparison around the circular region of keypoint. Furthermore, as BRISK results into binary descriptor, the matching step can also be performed efficiently using the Hamming distance. Details on keypoint detection and description are provided below.

Similar to SURF, BRISK is modular and it is possible to use any keypoint detection with BRISK descriptor, or any descriptor with BRISK keypoint detector.

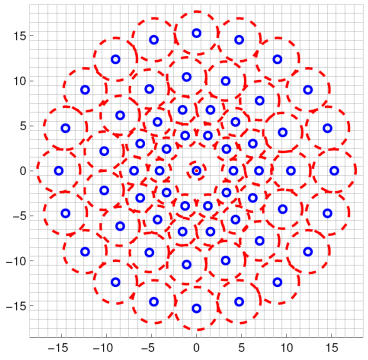


Figure 3.4: The BRISK sampling pattern [2]

3.5.1 Keypoint Detection

In principle, the BRISK keypoint detector is based on an extension of FAST corner/keypoint detector [67] and used AGAST [66] (which is very efficient) for detecting regions of interest in an image. However, FAST and AGAST are not scale invariant. To achieve scale invariance, BRISK also use scale-space pyramid using the FAST score V (see equation 3.4). To achieve true scale for each keypoint, BRISK uses continuous scale-space rather than discretized scale space (like SURF) [2].

To build scale-space pyramid, BRISK uses inter and intra octaves. Each inter octave is formed by sub-sampling the original image. Original image is downsampled by the factor of 1.5 to get the first intra-octave. Successive half sampling generates the rest. To detect keypoints, BRISK uses 9-16 mask which needs 9 consecutive pixels in a circular region of 16 pixels around the center pixel. First, the FAST keypoint detection is performed on each octave using 9-16 masks. Then, non-maximum suppression is performed in scale space using the criteria that the detection score V in both above and below layer needs to be lower than the current layer score. Detection and search of maxima in scale space for first octave is a special case, where the condition having the lower score in both above and below layer is changed to only above layer [2].

For continuous scale refinement, 2D quadratic fitting is performed on the three score-patches of detected maxima, one above, and one below. Next, a 1D parabola fitting is performed along the scale axis which results into final score and scale estimate [2].

3.5.2 Descriptor

Once keypoints are extracted from an image, the next step is to extract the descriptors. BRISK provides a rotation invariant binary descriptor, which is a binary string containing the results of simple brightness comparison test. To generate the binary string, BRISK uses a sampling pattern around a detected keypoint. Figure 3.4 shows the sampling pattern with $N = 60$ points used in BRISK. [2]

To build a descriptor that is rotation and scale invariant, rotated sampling pattern is applied around the keypoint. Note that BRIEF [65] descriptor is also constructed via brightness comparison, but BRISK uses a deterministic sampling pattern which results in a uniform sampling-point density around the keypoint. In addition, it also uses few sampling points, because single point is used in more comparisons, thereby reducing the complexity. As a result, a 512 bit descriptor is obtained which can be matched using the Hamming distance [2].

3.6 Fast Retina Keypoint (FREAK)

FREAK feature is another recently introduced part-based/local feature. Unlike SURF and SIFT, which provides both, keypoint detection and description, FREAK [3] is only a keypoint descriptor. Any other keypoint detector can be used in combination with FREAK. FREAK [3] is inspired from the human visual perception and provides binary local keypoint descriptor. Particularly, it is inspired from the retina, due to which it is referred to as Retina keypoint, whereas fast comes from the speed achieved by the binary nature of the descriptor. The main part, which is inspired form human visual system is the sampling grid used to compare pairs of pixel intensities. For instance, we can use random pairs (as in BRIEF [65] and ORB [9]) or circular pattern with equally spaced circles concentric (as in BRISK [2]). FREAK uses retinal sampling grid (which is also circular) where the density of points near the center are higher than the farther points [3]. The inspiration behind this sampling comes from retina where higher resolution is captured in the fovea (area near the center of the retina), whereas a lower resolution is captured in the perifoveal (outermost area of the retina). To realize this inspiration into FREAK, kernels of different sizes are used for every sample point [3]. The size of the Gaussian kernels is changed with respect to the log-polar retinal pattern. In addition, it includes redundancy that brings more discriminative power. Figure 3.5 shows the sampling pattern of FREAK.

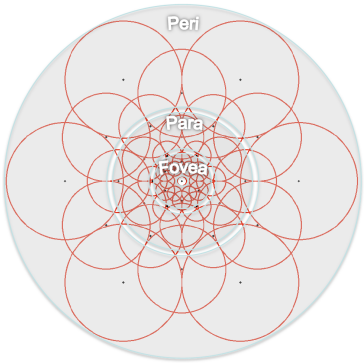


Figure 3.5: Sampling pattern used in FREAK [3]

A binary descriptor is constructed by thresholding the difference between selected pairs and corresponding Gaussian kernel (see equation 3.10).

$$F = \sum_{0 \leq a < N} 2^a T(P_a) \tag{3.10}$$

Where F is a descriptor, P_a is a pair and N is the desired size of the descriptor.

$$T(P_a) = \begin{cases} 1 & if(I(P_a^{r1}) - I(P_a^{r2})) > 0 \\ 0 & otherwise \end{cases} \tag{3.11}$$

$I(P_a^{r1})$ is smoothed intensity of the first field of the pair P_a and $I(P_a^{r2})$ is smoothed intensity of the second field of the pair P_a.

An important step while creating a FREAK descriptor is the selection of pairs which are not highly correlated and discriminant. This is achieved by coarse-to-fine ordering of the

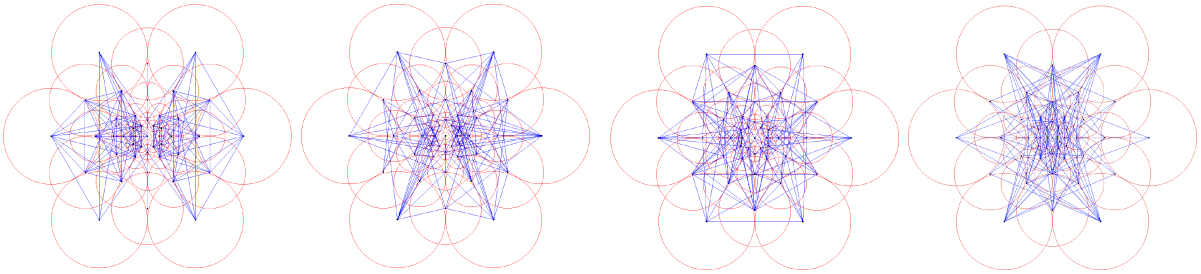


Figure 3.6: Coarse-to-fine sampling used in FREAK [3]

difference of Gaussian. Furthermore, saccadic search is used to parse and construct the descriptor in several steps. It starts by searching with the first 16 bytes of the FREAK descriptor (which is actually the coarse information) and if the distance is smaller than a threshold, the comparison is continued to the next bytes to analyze finer information. This process even accelerates the final matching step. In this process, more than 90% of the candidates are discarded at the first parsing step. The orientation estimation is performed using the sum of local gradients over selected pairs. In total 45 pairs are used to estimate the orientation.

4 Generic Framework for Information Segmentation: The Proposed Approach

4.1 Introduction

This chapter provides a detailed description of the presented generic framework for information segmentation. The generic nature of the presented framework makes it applicable to a variety of documents including technical drawings, magazines, administrative, scientific, and academic documents digitized using different methods (scanners, RGB cameras, and hyper-spectral imaging devices). A highlight of the presented framework is that it does not require large training sets, rather a few training samples (for instance, four) lead to a high performance. Several cases where this approach works better than previously existing methods are demonstrated in Chapters 5, 6, and 7. In addition, the presented framework is simple and can be adapted quickly to new problem domains.

The presented framework is based on part-based features. A part-based feature refers to local properties of a part/local-area/patch of an image [68]. These properties can be based on change in intensity, color, texture, gradient, etc. Local features are usually distinct from their neighborhood. For example, corners of an object, usually exhibits different properties than the object itself.

Part-based features perfectly suit to solving the problem of information segmentation [8, 58]. This is because a required piece of information can be located anywhere in a document image. Therefore, it is important to have a method that is able to first locate all important points which need to be segmented and then based on their characteristics, can classify them as different types of information. Part based approaches can

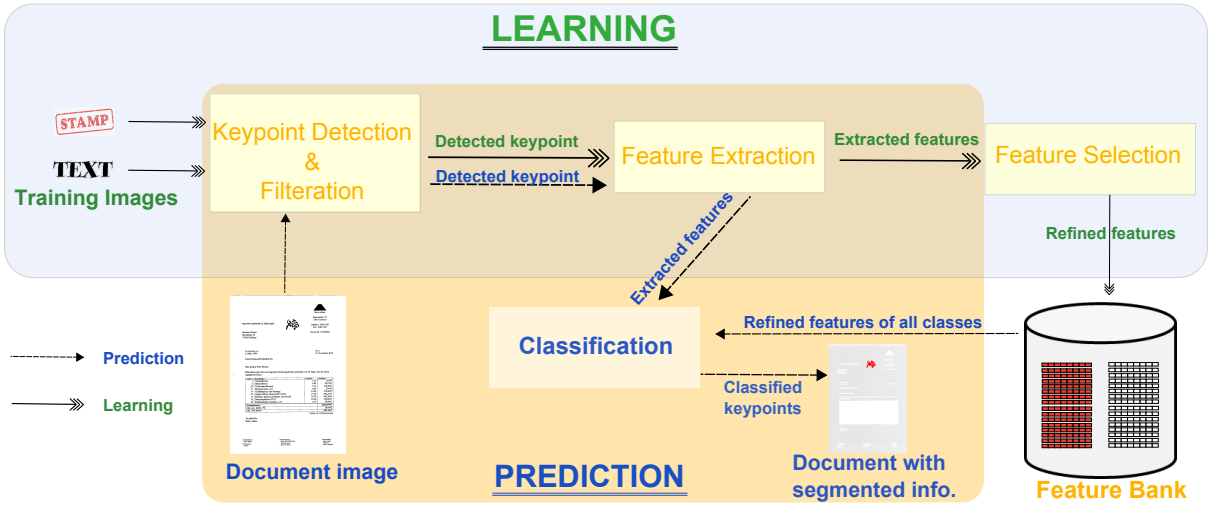


Figure 4.1: Workflow of information segmentation framework

serve both the purposes, i.e., identification of important areas, and feature extraction from these areas.

4.2 Framework

An overview of the presented information segmentation framework is illustrated in Figure 4.1. The presented framework is based on supervised learning and divided into learning and prediction. During the learning, the framework tries to learn various properties (e.g., image gradient or distribution of gray values) of different types of information that are to be segmented from a document image. To learn the properties of different types of information it is important to first have some sample images for each of the target information classes. Once these properties are learnt, feature banks containing features for each type information are generated. Later, the prediction uses these feature banks to segment different types of information available in previously unseen documents.

Note that the presented framework is not limited to a specific part-based features, and can be used in conjunction with any part-based features like, SURF [8], BRISK [2], ORB [9], FREAK [1]. Further a combination of different feature detection and extraction methods can be coupled with the presented framework. The choice of feature extraction is based on performance needs. If a system with high performance (in terms of accuracy) is needed, SURF is a better choice as it provides a very good description of an image patch based on image gradients. On the other hand, FREAK [3], ORB [9], and BRISK [2], are good

Detector	Detection time [sec]
FAST	0.01707
ORB	0.09065
BRISK	0.10756
SURF	1.14996

Table 4.1: Average keypoint detection time for different detectors

Descriptor	Extraction time [sec]
BRIEF	0.05557
ORB	0.07054
FREAK	0.11085
BRISK	0.04461
SURF	1.86130

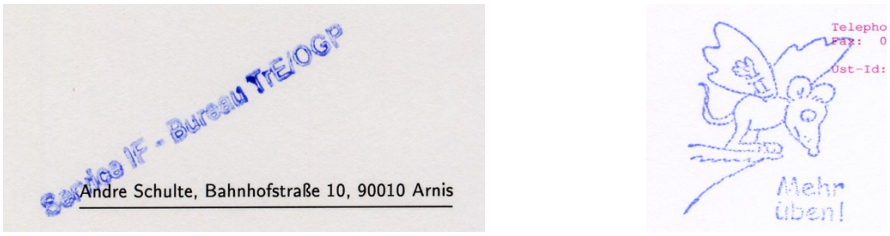
Table 4.2: Average feature extraction time

choices if efficiency is more important than accuracy in a particular application. This is because FREAK [3], BRIEF [65], ORB [9], and BRISK [2] are binary descriptors which are computed based on simple comparisons of different locations in the keypoint patch. Tables 4.1 and 4.2 provides details on detection and extraction time required by different detectors and extractors on a single image. More details on different part-based features, their detection, and extraction can be found in Chapter 3.

The framework consists of various important components, including keypoint detection, feature extraction, feature selection, and classification. Some of these components are used both in learning and prediction, while some are specific to each phase. The different components of the presented framework are described in detail below.

4.2.1 Keypoint Detection

A keypoint, also referred to as an interest point, is a spatial location in an image that contains important information and usually indicates a highly distinctive location in an image [68]. Keypoint detection is an important part of the presented segmentation framework. It is used both in the learning as well as in the prediction. During learning, keypoint detection is performed on samples (pre-segmented) images to locate important points in the sample images. During prediction, keypoint detection is performed on a complete document image in which information segmentation is needed. A keypoint can be found anywhere in a document, in different scales, as the same keypoint may appear



(a) (b)

Figure 4.2: Overlapping stamp with text

in images taken at different points in time at different resolutions and different rotations.

There are several possibilities to detect keypoints. One possibility is using connected components based approach to detect keypoints where each of the connected component is considered as a keypoint in an image [39, 43, 49]. This approach is very fast and simple. However, the main problem with connected component based approach appears when different types of information, e.g., stamp and text are overlapped with each other. In this case, the overlapping block will be considered as a single connected component, which is problematic for both learning and prediction. This is because a single component possesses the properties of different classes and using this in learning will increase the confusion between different classes. During prediction, the complete block will be classified into one class. This leads to problems in the segmentation of overlapping components which is a normal scenario in document images [28, 69] as depicted in Figure 4.2.

Another possibility is to use a sliding window approach as in other applications, e.g., object detection [70, 71], where each window is considered as a keypoint. The use of sliding window ensures that whole document is covered. However, the use of the sliding window approach remarkably reduces the overall performance (time) of the system due to its high computational cost. In addition, selection of appropriate window size is also an important parameter, which varies a lot based on the resolution of the images and on the type of information that is to be segmented.

In the presented framework, part-based keypoint detection techniques are used. Note that, regardless of the later used feature description, any keypoint detection technique can be used, e.g., FAST [1], BRISK [2], SURF [8], Harris corner detection [64]. However, based on different experiments, SURF keypoint detection mechanism proved to be the best option. SURF keypoint detection is based on scale space representation, where the keypoints are looked for both in scale and space. In addition, SURF is very stable and

invariant to scale, space, and rotation. Furthermore, SURF uses integral images, which increase overall efficiency of the whole process both in the detection and description steps. More details on SURF can be found in Section 3.1. Once keypoints are detected, the next step is to filter out irrelevant/unimportant keypoints. This step is necessary because, a small black point on white background is also a keypoint. However, depending upon the application, it needs to be decided which of the detected keypoints should be considered as relevant. In the presented framework, this decision is made on the detection score of each keypoint. In case of SURF, each of the detected keypoints has a Hessian response score, which shows how important a detected keypoint, is in terms of the Hessian response. The higher the Hessian response, the important the keypoint is. The Hessian response score is generally used to filter out irrelevant points. In the context of the presented generic framework, a Hessian threshold of 400 is used to filter out irrelevant/unimportant keypoints. Increasing this threshold will reduce the number of points and only most important keypoints are left, while decreasing the threshold increases the number of detected keypoints, which will contain both important and potentially unimportant keypoints.

4.2.2 Feature Extraction

After detection and filtration of keypoints, the next step is to perform feature extraction. The feature extraction step tries to extract different properties which best describes the area defined by a keypoint. These features are then used as a representative of the keypoint area. The feature extraction is used in both the learning and prediction. In the learning, feature extraction is performed based on the detected keypoints from sample images. Whereas in the prediction, feature extraction is performed for the detected keypoints on a given document in which information is to be segmented. The aim of feature extraction is to have unique and robust descriptions of the detected keypoints in a way which encodes maximum properties present in the keypoint region and that encoding fulfills the criteria mentioned in equation 4.1 and 4.2

$$if\ K_1 = K_2\ then\ D_1 = D_2 \tag{4.1}$$

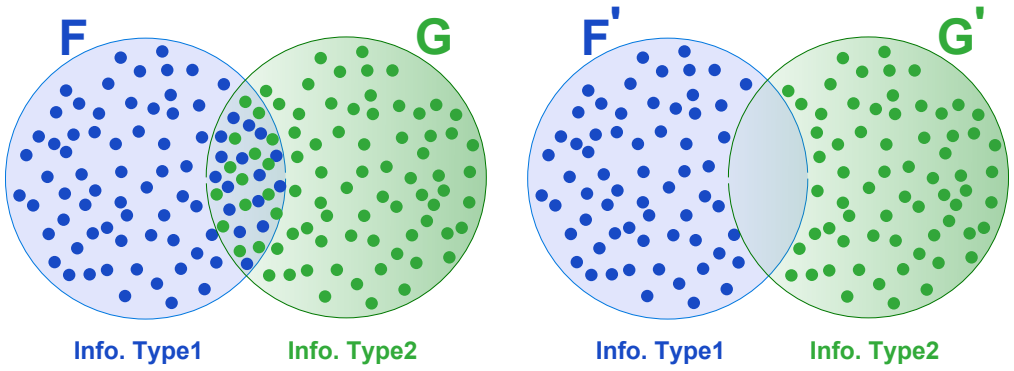
and

$$if\ K_1 \approx K_2\ then\ D_1 \approx D_2 \tag{4.2}$$

Where K_1 and K_2 are the detected keypoints in an image while D_1 and D_2 are the corresponding descriptors of keypoints. This means if the patch described by two keypoints is same, their encoding/description should also be same. Similarly, if the patch described by two keypoints is approximately same, their description should also be approximately same. A detailed performance analysis of different features on the problem of stamp segmentation are provided in Chapter 6.

4.2.3 Feature Selection

Once the detected keypoints are encoded by their respective descriptors, the next step is to perform feature selection. The purpose of this step is to select only the representative and relevant features, and ignore the features which are irrelevant and common among different types of information (common features among different classes cause affect classification [52–54]). The feature selection step is performed only during the learning and is based on the comparison of features in different classes, i.e., signatures, text, graphics, and stamps. The discriminative feature selection has already shown promising results for different problems, including object detection [53, 54], and recognition [52]. In addition to the work in literature, the experiments conducted through the course of this thesis also confirmed that feature selection helps in getting a high performance system with less training data [69, 72]. In addition to the work in literature, the experiments conducted through the course of this thesis also confirmed that feature selection helps in getting a high performance system with less training data [69, 72]. In the past, different sophisticated methods have been proposed for selecting relevant features from a given feature set. For instance, Naikal et al. [52] use sparse PCA to select the most discriminative features and Fürst et al. [54] use AdaBoost based feature selection. In the presented framework, the feature selection is based on inter class distance. Here, given sets of features 'F' and 'G' for two different classes, the feature set $F^{'}$ and $G^{'}$ are the selected discriminative features such that $F^{'} \subseteq F$ and $G^{'} \subseteq G$. The feature selection is performed using the feature evaluation function (equation 4.3), where inter class feature comparison is performed and common features among different classes are removed. Here θ represents an empirical threshold that can be adjusted according to a particular application of the said framework, for example, when separating signatures from text and when segmenting



(a) Features before filtration (b) Features after filtration

Figure 4.3: An example of feature selection

stamps from text.

$$Selection(F,G) = \begin{cases} 1 & dist(F,G) \geq \theta \\ 0 & dist(F,G) < \theta \end{cases} \tag{4.3}$$

A feature is retained if the evaluation of feature selection function returns 1. This means if two features from different classes are not similar, they will be retained in the corresponding feature set, otherwise they will be removed. The similarity between the features is measured using the Hamming (for binary descriptors) or the Euclidean (for floating point descriptors) distance. Figure 4.3 shows the features of two different information types before and after removal of similar features.

A separate feature bank is created for each of the targeted class at the end of feature selection (which is the last step in learning). These feature banks contain the most representative features for each class.

4.2.4 Classification

The classification step is an important part of prediction where each of the detected keypoint on a document image is classified into one of the targeted class based on its features. In the prediction, the feature banks of each of the targeted classes are used to finally classify each of the keypoints into one of the different types of information, i.e., stamp, signatures, or text. In this step, different supervised learning based classifiers can be used, including decision trees [73,74], Support Vector Machines [75], Boosting [76], etc.

In the presented framework, nearest neighbor classification is used to classify keypoints. The reason of using nearest neighbor classifier is that it does not require any training time and also have other various other favorable properties as pointed out by Boiman et al. [77]. In addition, the results based on nearest neighbor are easily explainable, which forms the basis for the systems which can classify and also explain their results. This explanation is very difficult in model-based classifiers, e.g., neural networks. It is also possible to use approximate nearest neighbor approach (a variant of the nearest neighbor), which is more efficient than nearest neighbor. However, our experiments showed that the use of approximate nearest neighbor results in performance drop. To classify each keypoint in a document image, its respective descriptor is compared with reference features of each information type from training data using nearest neighbor classifier that result in a distance for each class. Each keypoint is assigned to the class, which is at the minimum distance from this keypoint.

4.3 Applications

The presented framework is generic and can be used for multiple purposes. In this thesis it is applied to segment signature from scanned document images (see Chapter 5). In addition to scanned document images, the presented framework is also used to segment signatures from hyper-spectral document images (see Chapter 8), where it is not a 3 channel document, but a multi-dimensional e.g., 240 channel image containing response of the document for different wavelengths of lights. Forensic document examiners are already using hyper-spectral imaging to establish the authenticity of documents. Furthermore, stamps, another important seal of authenticity, are segmented using the presented information segmentation framework (see Chapter 6). Another application of the presented framework is on text/graphics segmentation in technical drawings, where it is important to segment text components, which are touching graphics (see Chapter 7). These are few applications; however, it can be used for segmentation of any type of information from document images.

Part I

SCANNED DOCUMENT ANALYSIS

5

Signature Segmentation from Administrative Document Images

Signatures are a widely used authentication mechanism in many industries such as banking and law [80]. In the last three decades, researchers have developed various offline (using only spatial information, such as scanned signature images) and online (using both spatial and temporal/dynamic information) signature verification systems. A common issue with nearly all of the existing signature verification systems is that they are built on the assumption that signatures are available pre-segmented or pre-extracted from document images. In addition, existing publicly available datasets for the development and evaluation of signature verification systems also contain mere signatures and not the complete documents containing signatures and other/or information. This chapter argues that these settings are not realistic and, in reality, experts encounter cases where signatures are written on documents having a lot of other information than just the signatures, e.g., machine-printed text, ruling lines, and logos. In these real world scenarios, due to the lack of segmentation, the existing signature verification and identification systems cannot be used "as-is".

To better assist forensic experts, a system should have the capability to automatically extract/segment signatures from documents like bank checks, forms, bills, wills, etc. This chapter focuses on the challenges faced by researchers when developing complete automatic document analysis systems capable of performing signature segmentation/extraction from documents and then performing signature verification. It is em-

[0]This chapter is an adapted version of the work presented in Ahmed et al. [78] "Signature segmentation from document images." In *ICFHR* 2012, and Ahmed et al. [79] "Extraction of Signatures from Document Images for Real World Applications", In *JASQDE* 2015".

phasized that automatic signature extraction/segmentation from document images is a relevant problem faced by forensic document examiners (FDEs) and compares the various approaches currently available. Section 5.1 provides an overview of the systems available for segmentation of signatures from document images. Section 5.2 presents an alternative method for extracting signatures from document images. The presented method is based on the generic information segmentation framework presented in Chapter 4 and is capable of distinguishing machine-printed text from signatures. Section 5.3 presents details about the dataset used for evaluation of the segmentation method. Section 5.4 reports the evaluation protocols and results of the proposed signature segmentation method. Section 5.5 discussed areas requiring further research and Section 5.6 concludes this chapter.

5.1 Existing Segmentation Systems

To date, there has been very little research in the area of segmentation and extraction of signatures from documents, especially from document images. However, some research has been undertaken that focused on the segmentation of handwritten text from machine-printed text. This section provide an overview of both cases, i.e., the existing systems available for segmentation of handwriting from machine-printed text, and those used for the segmentation of signatures from documents and bank checks.

5.1.1 Printed & Handwritten Text Segmentation

Imade et al. [81] proposed a method for segmentation and classification of printed characters, handwritten characters, photographs, and painted image regions using feed-forward neural networks. Similarly, Kuhnke et al. [82] proposed a classification system which reads a raster image of a character and outputs confidence values for machine-written and hand-written character classes. Features from machine-written and handwritten text are extracted and passed through a feed forward neural network to obtain the confidence score. Guo and Ma [83] addressed the problem of separating handwritten annotations from machine-printed text within a document. Here Hidden Markov Models (HMMs) were used to distinguish between machine-printed and handwritten materials. Zheng et al. [84] proposed a system for detection of machine-printed and handwritten text in

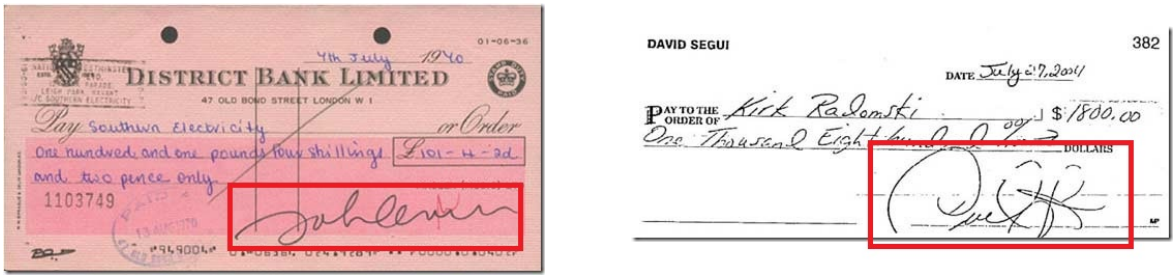
documents containing background noise. To achieve this, a Fisher classifier[1] initially separates the machine-printed and handwritten text from background noise. After this noise filtering process, Markov Random Field (MRF[2]) is used to segment the machine-printed text from handwritten text. Peng et al [86] used a two step approach to separate handwriting from machine-printed text using MRF. In the first step, patches/blocks of machine-printed, handwritten, and overlapped text (handwritten on machine printed) were extracted from the entire document using G-means. In the second step, MRF based relabeling was performed to separate overlapped text into machine-printed and handwritten text using shape context based pixel level features. Similarly, Chanda et al. [87] used a chain-code feature with a Support Vector Machine (SVM) classifier for segmentation of machine-printed text from handwritten text. Mozaffari and Bahar [88], Banerjee and Chaudhuri [89], and Banerjee [90] proposed systems for the segmentation of handwritten from machine-printed text in Farsi/Arabic and Bangla, respectively. These methods are based on connected component level features and base line profile. More recently, Awal et al. [91] proposed a method for handwritten and machine-printed text separation using pseudo-lines and contextual relabeling.

5.1.2 Signature Segmentation from Bank Checks and Document Images

A detailed description of existing methods for signature segmentation is provided in Chapter 2. This section provides a brief overview of different methods available for signature segmentation in bank checks and document images. Djeziri et al. [22] proposed a method based on filiformity criteria for differentiating between the objects from handwritten lines on bank checks. Madasu et al. [23] used a sliding window method to calculate entropy and to fit the window to signature blocks on bank checks. Sankari et al. [24] proposed an approach for the segmentation of signatures from check images using prior knowledge about the possible location of signatures in the Cartesian coordinate space. Note that in case of bank checks, prior information about the location of signatures is generally already available (see Figure 5.1) which makes the segmentation process comparatively easy.

[1]Fisher classifier is a well known linear discriminant classifier. For further details, please refer to `https://en.wikipedia.org/wiki/Linear_discriminant_analysis`

[2]MRF is a well profound graphical model for joint probability distribution. For further details, please refer to `https://engineering.purdue.edu/~bouman/publications/tutorials/mrf_tutorial/view.pdf` and [85]



(a) (b)

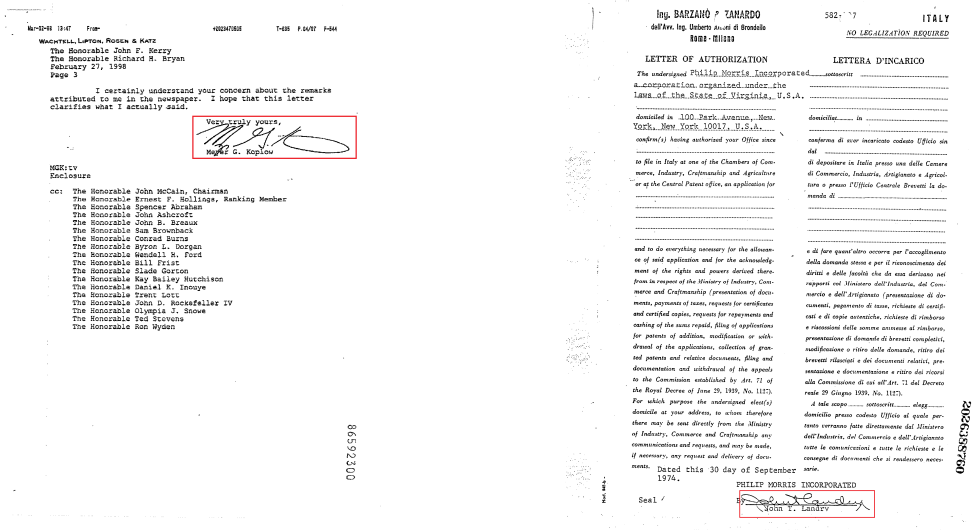
Figure 5.1: Bank check images([4])

Figure 5.2 shows documents apart from bank checks that also contain signatures and the prior information regarding the particular location/position of signatures on a document is not known. To deal with these situations, Zhu et al. [92] proposed a method for segmenting signatures from complete documents using a saliency map. In addition to a signature segmentation method, they also introduced a publicly available dataset called Tobacco-800 which consists of complex document images containing information about signatures on printed text documents. For further details on the Tobacco-800 dataset, please refer to Section 5.3. Mandal et al. [27] proposed an approach for signature segmentation using conditional random fields and reported results on a subset of the Tobacco-800 dataset, i.e., 105 images out of the total 1290 documents contained in the dataset. In another method, Mandal et al. [28] proposed an approach for segmentation of signature by also segmenting those characters which are touching the signature strokes. Estaban et al. [93] detected the position of signatures in document images using the accumulative evidence technique. However, this method needs a signature sample in an advance, which needs to be segmented.

Furthermore, some commercial systems are available. These systems are capable of finding one or two signatures in bank checks as well as (IRD[3]) images and snippets, and applying signature verification on these segmented signatures, e.g., SignatureXpert-2[4] by Parascript. The problem with commercial systems is that the data on which they are trained are never made publicly available. Also, details about how these systems work and the algorithms applied is proprietary information.

[3]An Image Replacement Document (IRD) is a replacement check on paper, generated from the electronic image of an authentic chapter check.

[4]http://www.parascript.com/recognition-products/forms-processing/signaturexpert-2



(a) (b)

Figure 5.2: Documents having signatures at different positions.

5.2 Part-Based Method for Signature Segmentation

This section provides an insight into the author's presented method for signature segmentation from document images. The presented method is based on the information segmentation framework introduced in Chapter 4. The method is based on local features, i.e., SURF. For each of the SURF key points, a 128 bit descriptor is extracted which represents that particular key point. This descriptor is used to find similarities between different parts of the image. For extraction of SURF, a Hessian threshold of 400 was used, i.e., all the key points having a Hessian threshold of less than 400 were ignored. This filtering removes unimportant features from the images. For particular details of SURF, please refer to the Section 3.1.

For training purposes, ten documents were used from the Tobaaco-800 dataset containing machine-printed text and signatures. To ensure the performance of the system, it is required that all the machine-printed text is separated from the signatures. As the Tobbaco-800 dataset does not include ground truth information for machine-printed text, two new images for each document were manually generated by dividing them into the printed text and signatures image components. These generated images were then used for training the system. Pixels containing overlaid areas were extracted for both the

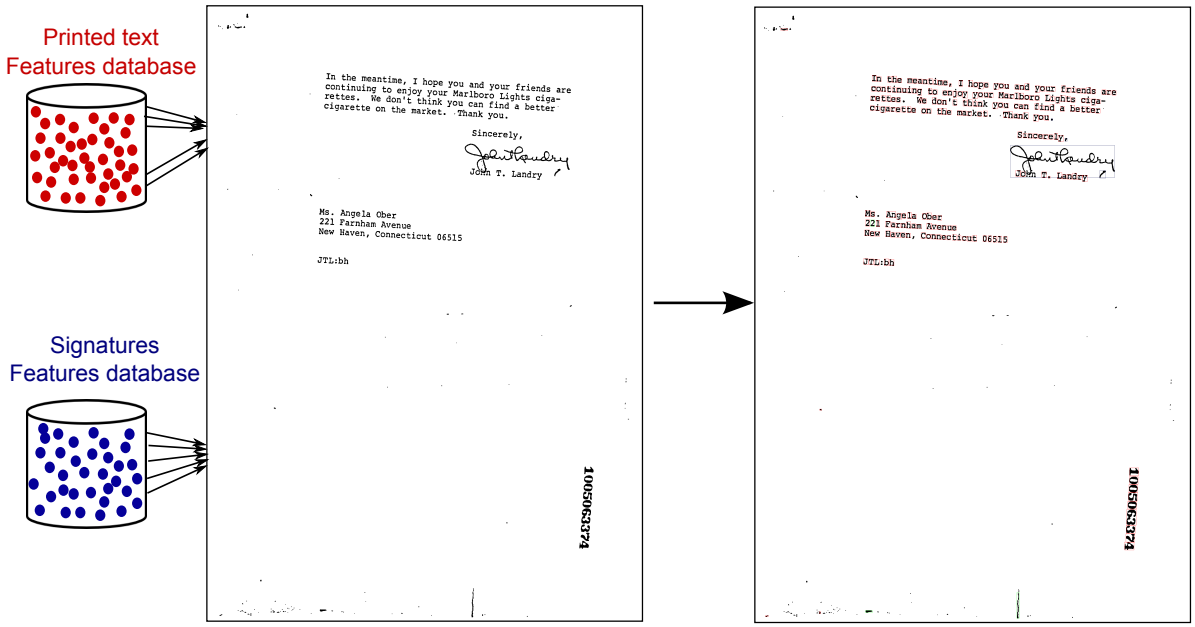


Figure 5.3: Extracted and marked connected components from question document image

printed text as well as signature image of the training set.

For all of the printed text components, the extracted printed text key points and their respective descriptors were added to a printed text features database. Similarly, for all of the signature components, the extracted signature key points and their respective descriptors were added to a signature features database. Feature selection was performed on the descriptors databases to remove similar features from both (printed and signatures) databases. The databases removal of similar features serve as the reference for matching the signature and printed-text features during testing. Figure 5.3 shows the training procedure used.

To segment a signature from a document containing both signatures as well as printed text, the connected components are extracted. Then the SURF features are extracted for each of the connected component (full signature component, full printed text component, and the overlapping signature and printed text components). The descriptors of the key points are then compared with the descriptors of printed text keypoints and signature keypoints from the two reference databases. The Euclidean distance metric is used as a distance measure.

Finally, for the classification of connected components, a majority voting approach is

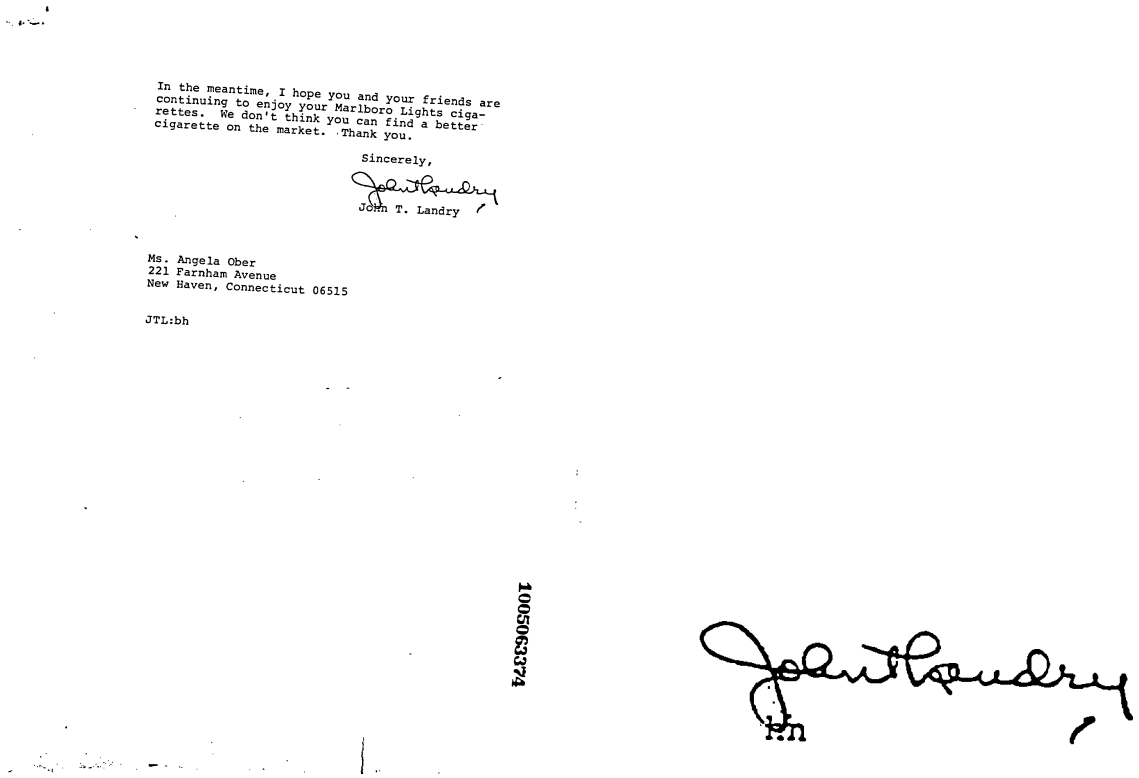
applied. If a connected component's keypoint has less Euclidean distance to the signature keypoints reference database as compared to the printed text keypoints reference database, one vote is added to the signatures class and vice versa. The process is repeated until all of the keypoints of a connected component are assigned to one of the two classes. A connected component is assigned to one of the two classes based on majority voting of its keypoints (See Figure 5.3). In case a signature is overlapping with the printed text, the whole overlapping part will be detected as a single connected component. If the connected component is partially overlapping with text, most of the part of overlapping connected component usually is signature and only some text components, which are touching the signature. In that case, the component is classified as signature, as it has more number of keypoint from signature than printed text (see Figure 5.4b). Once all of the connected components are marked as printed text or signature, separate images for signatures are generated. To segment the signature from the test document, the original image is cloned and bounding boxes of all connected components of printed text are filled with white color on that image, which in turn results in a segmented signature image.

As a post processing step, horizontal run length smearing is performed on the segmented signature image. Applying smearing merges all of the neighboring components. Connected components are extracted from smeared images and all of the small connected components are neglected. The remaining components are considered as signature patches. Figure 5.4 shows the extracted signatures from the document shown in Figure 5.4(a). One of the main advantages of our approach is that it requires a very limited number of training samples.

5.3 Dataset

Currently, to the best of the author's knowledge, there are two publicly available datasets that contain information about signature patches/zones. These are the Tobbaco-800 dataset [25] and the Maryland Arabic dataset [94]. The Tobbaco-800 dataset contains 1290 images containing the handwritten and machine-printed text in English as well as machine-printed logos. There are 900 labeled signatures in this dataset. The Maryland Arabic dataset contains 169 images containing handwritten English and Arabic text along with 149 labeled signatures.

To generate results comparable to the other approaches, such as those undertaken by Zhu



(a)

Figure 5.4: An administrative document image(a) and the extracted signature (b)

et al. [25], we performed evaluations of our presented system on the Tobbaco-800 dataset. This dataset contains only the ground truth information about the logos and signatures contained in a document on the patch level (i.e., which block in the image contains the signature and which block the logos). However, if the signature is overlapping printed text in that patch, only general information about the patch is supplied. As a result, it is not possible to evaluate the individual pixels in the image to determine which pixels are a part of signature and which are part of the printed text. The datasets currently available only have the patch level ground truth information about signatures available. For better investigations, a pixel-level ground-truth would be preferable.

To compare the presented method with the method proposed by Mandal et al. [95], we have used a subset of images from the Tobbaco-800 dataset containing only machine-printed text and signatures.

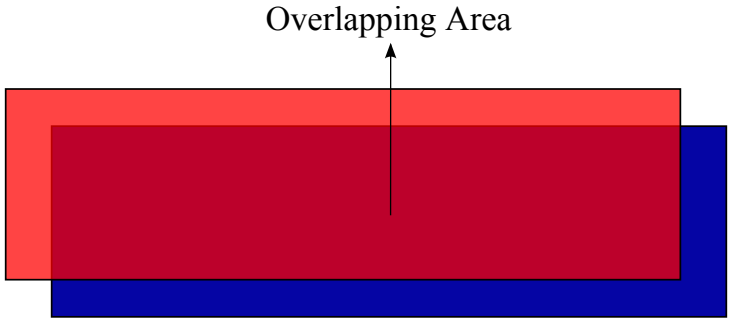
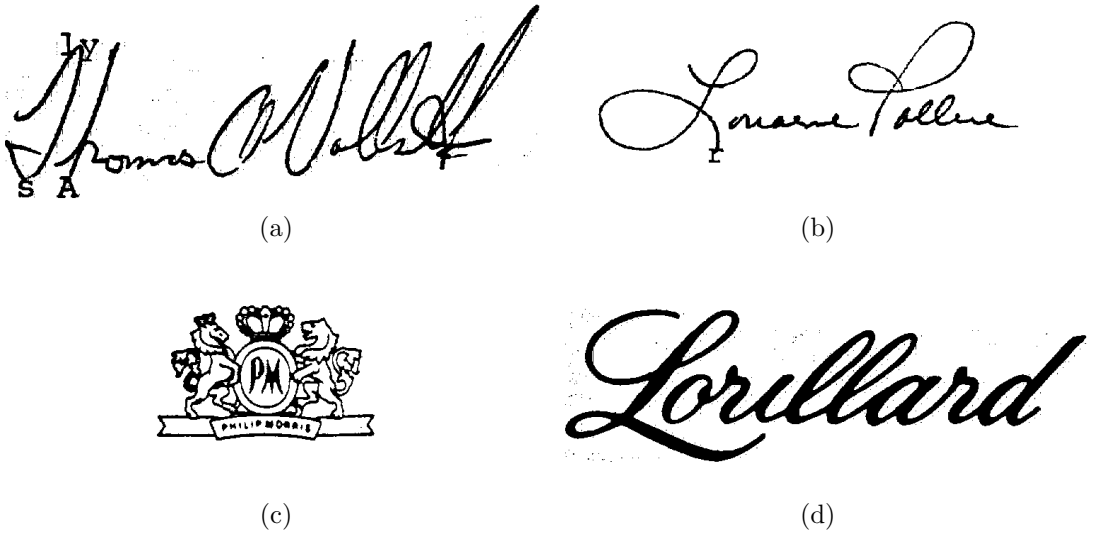


Figure 5.5: Overlapping area between ground truth (RED) and detected (BLUE) signature patch



(a) (b)

(c) (d)

Figure 5.6: Examples of correctly segmented signatures (a,b) and false positives (c,d)

5.4 Evaluation

To evaluate the performance of the presented signature segmentation method the precision and recall measures are used. As mentioned in Section 5.3, the ground truth of the available dataset contains only patch level information of the signatures. Therefore, the precision and recall is also calculated on the patch level information for the images. The signature is considered detected if there is at least a 40% of overlap between the ground truth and the detected signature patch.

The evaluation results of the proposed method are presented in Table 5.1. This method has a recall rate of 100%, which means that all the signatures were extracted successfully.

Table 5.1: Signature segmentation results on patch level

Method	Precision%	Recall%
Proposed method[5]	56.52	100
Mandal et al.(105 images) [95]	not reported by authors	98.56
Guangyu et al. [25, 96]	not reported by authors	92.8

A minor drawback of this method is, however, that the precision is currently quite low. One reason for this is that the images containing logos were also tested. The presented method sometimes incorrectly marked logos as signature patches during the segmentation process. However this drawback can be removed by adding the class "logo".

Figure 5.6 shows some of the segmentation results of the proposed method. Qualitatively, the correctly segmented signatures are comparable to manually cropped signatures. Figure 5.6 also shows some examples of false positives.

As can be seen, the presented method preforms quite well on a difficult subset. More than every second extracted patch is a signature, with all the signatures contained in the images successfully extracted. This outcome is very positive suggesting it will be a useful technique for document examiners.

5.5 Scope for Future Research

In order to automate the complete document analysis process, various methods need to be developed to extract different types of information from documents in a format usable for different applications. Likewise, reliable automatic segmentation methods that can be integrated with signature verification systems need to be developed to make it a viable resource for document examiners. To develop such methods, benchmark datasets on which signature segmentation systems can be evaluated in terms of recall (how many signatures/items are extracted from the document), precision (how many of the extracted items are actually signatures) and efficiency (both in terms of speed and complexity), need to be compiled. As discussed in Section 5.1, currently most of the existing signature segmentation systems are evaluated on subsets of Tobacco-800 dataset and (to the best of authors knowledge) there is no publicly available dataset specifically designed for signature segmentation. As discussed, the major disadvantage with the Tobacco-800 dataset is that it only contains patch level information about signatures, i.e., which block contains primarily signatures (meaning that some other information may also be there).

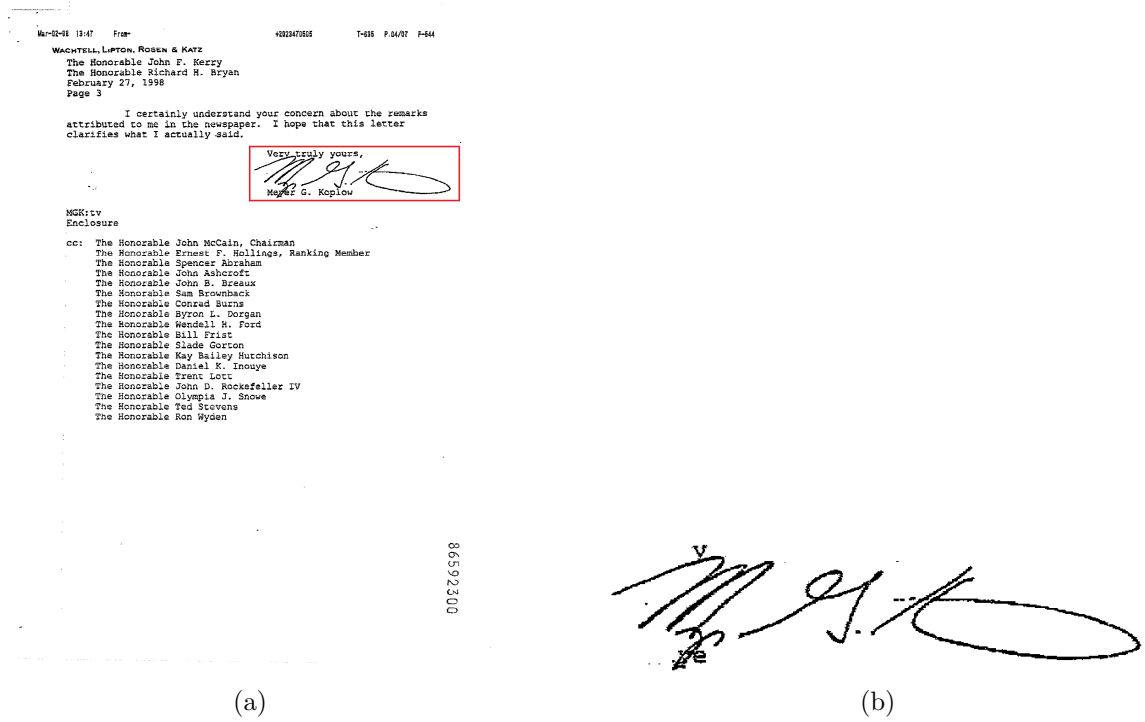
This non-availability of datasets shows the current lack of research interest in signature segmentation. Since 2011, some researchers have considered this problem but still there is a lot more that needs to be done in this area.

The authors are currently working on developing a large dataset for signature segmentation and verification. Along with patch level information, this dataset will have also signature stroke information (i.e., information about each pixel that belongs to signatures or printed text) and would be usable for testing complete signature segmentation and verification frameworks for analysis of documents containing signatures. Furthermore, in this dataset multi-spectral information about signatures and other parts of documents will also be incorporated.

In addition to developing datasets specific to signature segmentation, another important area requiring research is in the development of a system capable of performing layout-free segmentation of signatures. As previously discussed, signatures are not always located in the same place on a document (as shown in Figure 5.2). This means that an automatic system needs to be capable of finding signatures without using any prior information about the layout of a document and probable location of a signature. Some efforts to achieve this have already been made by Mandal et al. [27, 28] but there is still a lot more to be done in terms of quality of extraction, precision, recall, and efficiency.

Another important aspect is to tune signature verification systems in such a way so that they are capable of distinguishing between genuine and forged signatures, even in the presence of some noise in signatures, e.g., overlapping characters or missing parts of signature. The existing signature verification systems assume that the questioned signature image contains no other information than the signature itself, which is not always the case (see Figure 5.7 for reference).

Figure 5.7 (b) shows a very common scenario where most of the existing signature verification systems will misclassify this signature as a forgery simply due to the presence of text in the signature image. This extraneous text is considered as a part of signature during verification by current automatic systems. Therefore, further development is required to tune the existing signature verification systems so as to make them more robust to noise and touching components. A probable solution could be to use only parts of signature for verification rather than using the complete signatures [97, 98]. This, however, will raise further questions about which parts of signatures are to be used and why? In addition, a possibility is to use hyper-spectral imaging technology for distinguishing the signatures



(a) (b)

Figure 5.7: Example of signatures overlaying printed text.

from background noise. Hyper-spectral imaging provides wavelength information about the underlying text/signatures represented from visible to near infrared regions. This information may allow the program to very accurately segment the signature from its surrounding, even at pixel level [99].

5.6 Conclusions and Future Work

In this chapter, state-of-the-art automatic signature segmentation/extraction methods with the potential to be integrated with verification to perform authentication were discussed. The chapter outlined the limitations of the currently available automatic signature segmentation systems in real world scenarios. The main drawback of the current systems is that documents often contain information other than just the signatures, e.g., background text, lines, and logos. In order to authenticate these documents by performing signature verification, the signature must firstly be segmented and extracted from the document. The chapter assesses the various approaches that have been proposed for signature segmentation from document images and outlines their limitations. The chapter also notes the current lack of databases suitable for the development and test-

ing of complete document authentication systems involving signature segmentation and verification.

Furthermore, the author have presented a part-based method for the extraction of signatures from documents based on the SURF key points method. The presented method requires very limited training as compared to other contemporary approaches discussed in this chapter. The experiments were performed on the Tobbaco-800 dataset where all of the available signatures were successfully extracted. However, some false positives were also detected mostly due to the presence of other class information in the images such as logos. This limitation can be overcome by extending the method to identify other classes. By incorporating these modifications into the system, the precision of the method will be vastly increased. To remove the text components touching signature strokes, the method presented in Chapter 7 can be used.

It is anticipated that the author's dataset containing signature patch and stroke level information will be completed in the near future. Once completed, it will be made publicly available. Emphasis will be given to incorporating examples of signatures overlapping text and graphics in the dataset to make it more reflective of real world situations. The patch information will be recorded in both the visible and near Infrared spectra, as recommended by Khan et al. [99].

6

Stamp Segmentation in Administrative Documents

A stamp, similar to signature, is also considered as a seal of authenticity for documents [100]. Almost all official documents, e.g., financial, governmental, bank documents, checks, and even utility bills, are sealed with stamps and/or signatures. Furthermore, in various cases it is required to ensure authenticity of documents (to check whether they are genuine or not) before integrating them into the normal workflow of any business process, e.g., before issuing cash on bank checks, before releasing the claimed amount on invoices received by insurance companies, etc. In the past, researchers have developed various stamp verification systems. A common problem with nearly all of the existing stamp verification systems is that they are built on an assumption that stamps are already pre-segmented. This chapter argues that these settings are not realistic, as stamp is usually a part of document that contains other information as well (apart from stamps). Note that the location of stamps is usually more arbitrary than signatures as stamps can occur at different angles and at different locations based on the content of a document and the person sealing a document. Furthermore, stamps could belong to different categories, e.g., graphical, textual, regular and irregular shaped (see Figure 6.1). It is, therefore, required to first segment these seals (stamps) from documents in order to process the documents automatically.

Since the format of stamps may vary from organization to organization and even from person to person, it is not feasible to take all different templates for training and segmenting these stamps from documents. However, an important observation about stamps is that they usually exhibit different characteristics from other information, such as text,

[0]This chapter is an adapted version of the work presented in Ahmed et al. [72] "A Generic Method for Stamp Segmentation Using Part-Based Features" In *ICDAR* 2013

Figure 6.1: Stamps of different categories

present in documents. These characteristics usually include differences in color, ink distribution, size, and shape. Based on these differences, humans are also able to distinguish them. In the past, different methods have been proposed for stamp segmentation, which usually work by analyzing one or more of these characteristics. A detailed overview of existing stamp segmentation methods is provided in Chapter 2. In almost all of the existing approaches, it is assumed that stamps are colored objects and/or of some specific shape, and promising results are achieved. However, this is not always the case as black stamps are also in common use and scans of documents are often available grayscale only. Moreover, shapes of stamps also vary from organization to organization.

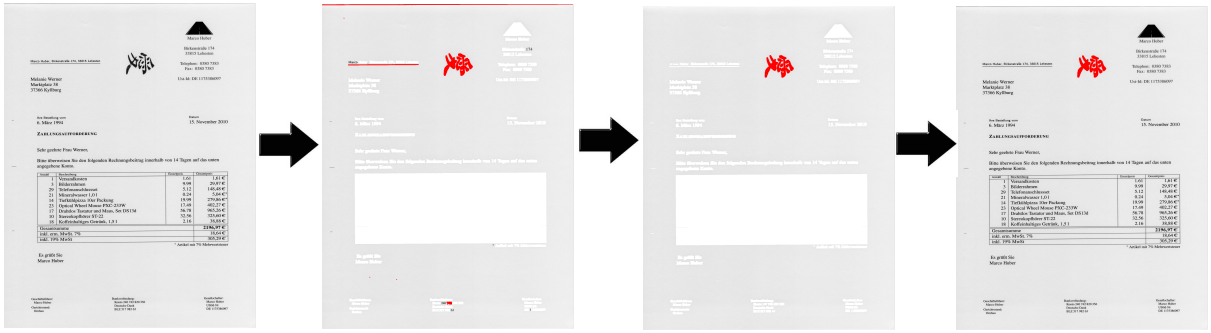
This chapter presents a generic, novel method for segmentation of stamps from document images. The presented method is based on the generic information segmentation framework presented in Chapter 4. The method is generic in a way that it can segment single/multicolored, monochrome, and even black stamps. In addition, it is capable of segmenting unseen stamps and stamps of any arbitrary shapes. Section 6.1 describes the methodology of the presented stamp segmentation method. Section 6.2 presents an evaluation of the presented method. Finally Section 6.3 concludes this chapter.

6.1 Part-Based Method for Stamp Segmentation

The presented stamp segmentation method is based on a combination of geometric features and the information segmentation framework presented in Chapter 4 . The part-based features are used to extract stamp candidates while simple geometrical features are used to filter out non-stamp objects from the extracted candidates.

To use the information segmentation framework, presented in Chapter 4, it is required to select appropriate keypoint detector and descriptor. In the presented stamp segmen-

tation method, FAST [1] is used as the keypoint detector to detect regions of interest in document images. The FAST keypoint detector is computationally efficient in comparison to other well known keypoint detection methods, e.g., SIFT [101], Harris [64], and SURF [8]. In addition, FAST gives a strong response on the edges, which makes it suitable for document images. Descriptor for each of the keypoints is computed using two different part-based descriptors are used, i.e., BRIEF [65] and ORB [9]. As their names suggest, both BRIEF and ORB are binary descriptors, which possess high discriminative power with a few bits. These descriptors are computed using simple intensity difference test around keypoints. BRIEF descriptor is sensitive to large rotations, whereas ORB is rotation invariant version of BRIEF. These descriptors are also computationally very efficient as compared to other local descriptors, e.g., SIFT [101] and SURF [8]. For more particular details about FAST, BRIEF, and ORB, please refer to Chapter 3.



(a) **Gray Scale Document Image**

(b) **Marked Stamp Components Using Part-based Features**

(c) **Marked Stamp Components after Geometric filtering**

(d) **Marked Detected Stamp**

Figure 6.2: Stamp segmentation steps

To detect stamps in document images, first a training set is created for stamp and non-stamp objects. The stamp training set contains a few stamps from each category, i.e., textual, graphical, regular shaped, and irregular shaped. It is important to mention that our training set does not include all the stamps, it just needs a few samples from each category. In the non-stamp training set images with text, logos, and tables are included. For each image in the stamp training set, connected components are extracted. For each bounding box of the connected components, keypoints are then extracted using FAST [1] keypoint detector. Furthermore, the descriptors are extracted for each of the detected keypoints using BRIEF [65] and ORB [9] feature descriptor methods, separately. In addition to the part-based features, simple geometric features, i.e., stamp bounding box height (SB_{height_i}), and width (SB_{width_i}), are also extracted for each connected component. Mean (μ_h and μ_w) and standard deviation (σ_h and σ_w) from all stamp bounding boxes

heights(SB_{height}) and widths(SB_{width}) are calculated using Eqs. 6.1, 6.3 and Eqs. 6.2, 6.4 respectively, where N is the number of stamp bounding boxes in training set.

$$\mu_h = \frac{\sum_{i=1}^{N} SB_{height_i}}{N} \tag{6.1}$$

$$\sigma_h = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(SB_{height_i} - \mu_h)} \tag{6.2}$$

$$\mu_w = \frac{\sum_{i=1}^{n} SB_{width_i}}{N} \tag{6.3}$$

$$\sigma_w = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(SB_{width_i} - \mu_w)} \tag{6.4}$$

All of these descriptors and the range for each geometric feature is saved in a database. The same process is applied to the non-stamp training set. Finally, we have a feature bank each for stamps and non-stamp components separately. It is to be noted that no vocabulary is constructed for these features as each feature is of importance here. Therefore, these feature banks contain all of the extracted features. In addition, no feature selection is performed and this is the reason to include geometric features.

To detect a stamp in a query image, it is first converted into gray scale (Figure 6.2 (a)). This conversion is done to make sure that no color information will be used during the further analysis. This image is then binarized to extract connected components. In the first step, keypoints for each bounding box are detected using the FAST keypoint detector. Furthermore, descriptor for each detected keypoint is extracted using BRIEF and ORB local descriptors. All the descriptors of the connected components from the query image are then compared to stamp and non-stamp descriptors from the database.

As the descriptors extracted using BRIEF and ORB are binary, therefore the Hamming distance is used for comparison of descriptors extracted from the bounding boxes and the descriptors available in the bags-of-features. The use of Hamming distance in-turn makes it computationally more efficient, as it can be computed using a simple XOR operation on bit level. A component is then classified as stamp or non-stamp based on majority voting. This means, if the number of descriptors similar to stamps is greater, then an

object is marked as a stamp, otherwise as a non-stamp component.

All of the objects which are detected as non-stamps are removed from the query image. Thereby, we are left with candidate stamp objects (with stamps and some misclassified components only). Figure 6.2 (b) shows the query image after removing non-stamp objects and marked stamp objects. This miss-classification occurs because, stamp training set also contains textual stamps, which are nothing but characters. Therefore, some descriptors in the stamp database are similar to descriptors in the non-stamp database. For example, features for logos are very similar to graphical stamps, which cause miss-classification of logos as stamps. As there is no feature selection performed, therefore, further filtering is required to filter out false positives.

To filter out the false positive from the stamp candidate components, geometric features based filtering is performed. For all of the candidate components (QB), the geometric features' values are computed (QB_{height} and QB_{width}). A component not satisfying the conditions in Eqs. 6.5 and 6.6 is removed from the candidate list.

$$f(QB_{height_i} = \begin{cases} stamp & (\mu_h - 3\sigma_h) \leq QB_{height_i} \leq (\mu_h + 3\sigma_h) \\ nonstamp & otherwise \end{cases} \tag{6.5}$$

$$f(QB_{width_i} \begin{cases} stamp & (\mu_w - 3\sigma_h) \leq QB_{width_i} \leq (\mu_h + 3\sigma_w) \\ nonstamp & otherwise \end{cases} \tag{6.6}$$

Figure 6.2 (c) shows the image after filtering out non-stamp objects using geometric features. Finally, the remaining components are referred to as stamp components, and regions for these components can be extracted from the original image. Figure 6.2 (d) shows the query image where a detected stamp is marked.

6.2 Evaluation

(a) (b) (c)

Figure 6.3: Extracted, ground truth, and overlapped stamps

6.2.1 Dataset

The presented method is evaluated on a publicly available dataset (StaVer[1]) for stamp detection and verification [19]. This dataset contains 400 scanned document images. Out of these 400 documents, 80 documents contain black stamps whereas the remaining 320 documents contain colored stamps. All of these document images are available in 200, 300, and 600 dpi. For each image, two different types of ground truths are available. One contains the pixel level ground truth, which means all of the pixels which belong to stamps are marked in the image. The other ground truth format contains bounding box information for each stamp. Hence, this dataset can be used for both pixel level as well as patch level evaluation of stamp detection. In addition, it contains different types of stamps ranging from rectangular, oval, to irregular shaped, and most importantly, textual stamps.

For evaluation of the presented approach, the training set is generated by using 36 documents out of 400. Out of these 36 training documents, only 6 contains black stamps whereas the remaining 30 are with colored stamps. Testing is performed on the remaining 364 documents (74 documents with black stamps, 290 documents with colored stamps). All the results are reported for 200 dpi documents.

6.2.2 Evaluation Protocol

In this chapter, the pixel level evaluation is used for reporting results as it is more realistic than the patch level, especially in terms of recall. To get the pixel level results, we looked for the number of pixels in the ground truth image which correspond to the pixels in the detected stamp image. These common pixels are then divided by the total number of pixels in the ground truth image, which correspond to recall of the presented method in terms of pixels. Figure 6.3 shows (a) the detected, (b) the ground truth, and (c) the overlapping stamps, respectively.

Table 6.1: Evaluation results for black stamps

Detector	**Descriptor**	Recall	Precision
FAST	*BRIEF*	72	74
FAST	*ORB*	73	83
Micenkova et.al. [19]		Not Applicable	Not Applicable

[1]http://madm.dfki.de/downloads-ds-staver

Table 6.2: Evaluation results for color stamps

Detector	**Descriptor**	Recall	Precision
FAST	*BRIEF*	56	48
FAST	*ORB*	57	62
Micenkova et.al. [19]		82.7	82.8

6.2.3 Results and Discussion

Table 6.1 and Table 6.2 show the recall and precision of the presented stamp segmentation method for black and colored stamps, respectively. Table 6.1 shows that the presented method has good recall and precision in case of black stamps. It can be seen that the method for stamp detection by Micenkova and van Beusekom [19] is not applicable to these stamps, all of the processing is performed in YC_bC_r with an assumption that stamps are always colored objects and, therefore, clustered in different colors than that of text. However, this assumption does not hold true for black stamps and, therefore, existing methods are not applicable to black stamps.

Table 6.2 reveals that recall and precision of the presented method are low in comparison to the method of Micenkova and van Beusekom [19]. It is because the presented method is not using any color information, therefore it is difficult to separate especially those stamps which are severely overlapping with other components. Figure 6.4 shows some of the cases where the presented method is unable to detect stamps, as most of them are overlapping with neighboring non-stamp components. Whereas, in case of Micenkova and van Beusekom [19], color clustering is done for finding candidates for stamps, therefore, it is possible to segment overlapping stamps if they are of different colors than that of the non stamp components/text. In the presented method, it is possible to include color information, if available, to segment stamps. This will in turn increase recall and

Figure 6.4: Severely overlapping stamps missed by the presented approach

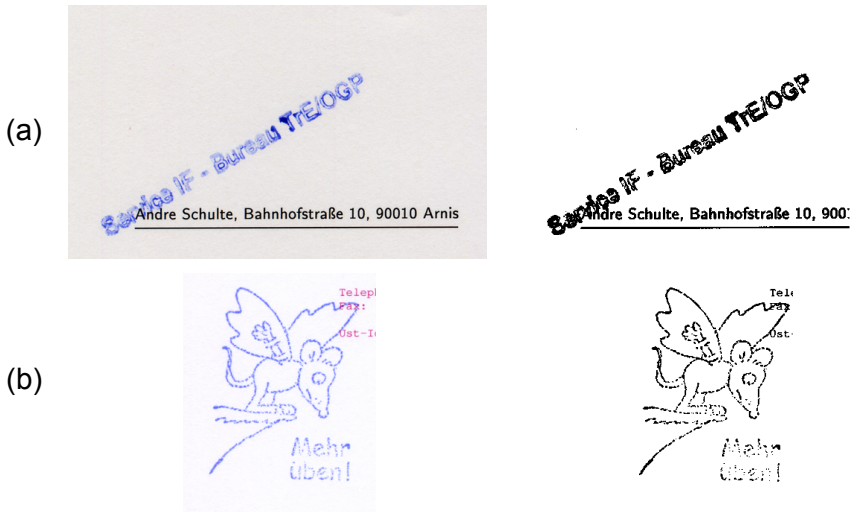


Figure 6.5: Partially overlapping stamps detected by the presented approach

precision of the presented method even in the presence of severely overlapping stamps. Nonetheless, if stamps are partially overlapping, the presented method is capable of detecting them. This is because, for partially overlapping objects, the number of votes in connected components are higher for stamp objects as compared to the non-stamp objects. Figure 6.5 shows some cases of overlapping stamps which were successfully segmented by the presented method.

Note that the training set for non-stamp objects, in the presented approach, also contain logos. However, some part-based as well as geometric features of non-stamp objects are more similar to graphical stamps and even in some cases to textual stamps. That is why, sometimes, logos are also marked as stamp components. The drop in precision is because of the presence of logos in documents. Figure 6.6 shows the cases where logos are marked as stamps. There are already part based methods available for logo detection. Therefore, precision can be improved by simply applying a logo detection method next in the hierarchy.

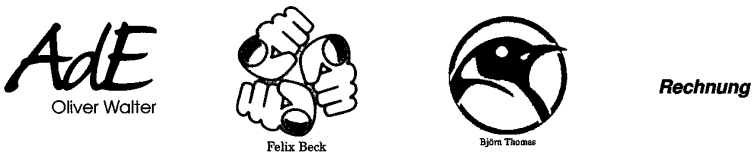


Figure 6.6: Misclassified objects as stamp objects due to similarity with graphical stamps

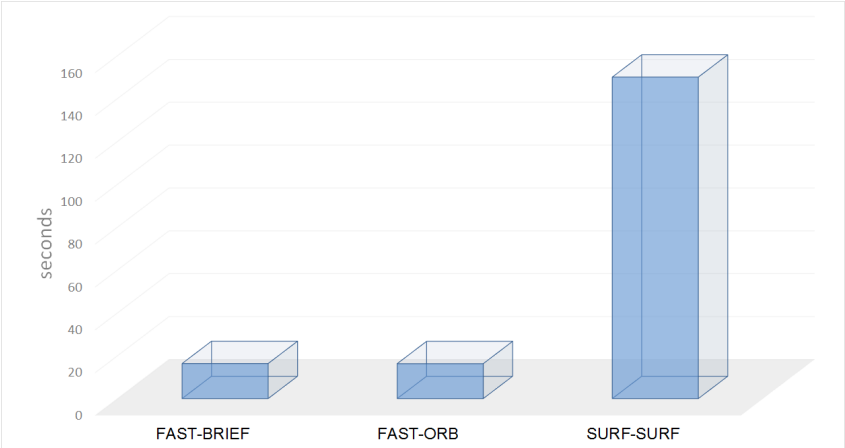


Figure 6.7: Computational analysis of different features

In terms of computational time, both BRIEF and ORB features in combination with FAST keypoint extractor and naive nearest neighbor classification require around 16.29 seconds to segment stamps from a document image of 200 dpi. To compare the computational difference between SURF and other descriptors, an experiment is performed where SURF is used as keypoint detector as well as descriptor. Figure 6.7 shows the results of this experiment which reveals that, it took more than 150 seconds on average to segment stamps when using SURF (both for keypoint detection and description). This shows that using SURF as keypoint detector is computationally very expensive in comparison to using FAST in combination with other binary descriptors.

6.3 Conclusions and Future Work

A generic and novel method for stamp detection is presented in this chapter. The presented method is based on a combination of part-based and geometrical features. It is able to segment stamps of all categories, i.e., textual, graphical, regular shaped, and irregular shaped. In addition, the presented method is able to detect colored as well as black stamps. The main highlight of the presented method is segmentation/extraction of black stamps, as the existing methods for stamp detection are not able to segment black stamps since they are based on the assumption that stamps are colored objects. We have achieved recall and precision of 75% and 84%, respectively.

In the future, it is planned to integrate logo detection with the currently proposed method, to further increase precision. Furthermore, other geometric features can be integrated

to increase recall of the system. To overcome the problem of overlapping stamps, the method presented in in Chapter 7 can also be investigated. Furthermore, the presented method is orthogonal to the method presented by Micenkova and van Beusekom [19] - so both methods can be easily combined to get the benefits of both, i.e., first color-based segmentation and then part-based processing.

7

Segmentation of Text Touching Graphics in Technical Drawings

Text/graphics segmentation is considered as an important step in the analysis of technical drawings (floorplans, maps, circuit diagrams). The aim of text/graphics segmentation is to extract two separate layers, one containing only graphical information and the other containing only textual information. Most of the existing methods focus on the separation of text components, which are non-overlapping with graphics. However, in technical drawings, different parts of text generally overlap with graphics and extraction of such overlapping text is an important challenge. This chapter describes a novel method for extracting text components especially the ones touching graphics in technical drawings. This method is based on the generic framework present in Chapter 4. The rest of the chapter is organized as follows. Section 7.1 summarizes the work related to text/graphics segmentation in general and extraction of touching text characters in particular. Section 7.2 provides an overview of the presented segmentation method. The presented method is evaluated on a publicly available dataset of architectural floor plans. Section 7.3 presents the evaluation results of the presented segmentation method. Section 7.4, finally, concludes the chapter and gives an overview for the future work.

[0]This chapter is an adapted version of the work presented in Ahmed et al. [49] "Extraction of Text Touching Graphics Using SURF" In *DAS* 2012

7.1 Related Work

The detailed description of general text/graphics segmentation method is provided in Section 2.3. This Section provides a brief overview of different methods available for text/graphics segmentation especially in technical drawings.

Fletcher and Kasturi [39] presented a method to extract text strings from mixed text/graphics images of technical drawings. This method is based on connected component analysis and works fine with non-touching text. Dori and Wenyin [40] performed a vector-based segmentation of text connected to graphics in engineering drawings. This method also focuses on touching characters using heuristics. Cao and Tan [41] presented a text/graphics separation method for overlapping text and graphics in map images. Adam et al. [102] used Mellin Fourier Transform too classify characters and symbols drawn on technical drawings. Most of the touching characters were successfully extracted by this method. However, a major disadvantage of this method is that it is very time consuming. Tombre et al. [43] proposed an improvement for the method proposed by Fletcher and Kasturi [39] by introducing some additional filters to apply on connected components. This method improved the results, but still some touching characters which are either at start or at the end of words were marked as graphical components. Roy et al. [44] further improved the approach of Tombre et al. [43] and used color information to separate touching text from the graphics. However, the main assumption of the method is that the color of text and graphics is different. Raveaux et al. [46] also used color information coupled with a graph representation. A basic assumption of the method is that the text is not touching graphical components.

Roy et al. [47] used the SIFT features and shape models with SVM for extraction of text touching graphics in geographical maps images. Hoang and Tabbone [48] introduced an approach that is based on the sparse representation framework and two appropriately chosen discriminative dictionaries one for text and other for graphics. This method is capable of extracting some of the touching text as well. Ahmed et al. [49] proposed a method for text/graphics segmentation in architectural floor plans. This method extends the method proposed by Tombre et al. [43] by providing a mechanism to calculate different thresholds dynamically. This method has good accuracy and is able to extract most of the overlapping text, but can only be used for architectural floor plan images. Do et al. [50] presented a method to extract text areas from graphical documents using sparse representation and multi-learned dictionaries (for both text and graphics).

(a) (b)

Figure 7.1: SURF features of text and non-text component

Mello and Machado [51] presented a method for text segmentation in vintage floor plans and topographic map images. The presented method is based on background removal, thresholding, histogram equalization, connected component analysis, lines removal, noise removal, and restoration.

7.2 Part-Based Method for Touching Text Segmentation

The presented method for extraction of touching text components is based on the information segmentation framework presented in Chapter 4. As the presented framework is based on part-based features, SURF is used to extract touching characters. It extracts the key points/points of interest from an image. Then each key point is represented by a 64/128 bit discriminative descriptor. Figure 7.1 shows key points extracted by SURF for text and non-text images. Each of the extracted key points contains information about the x, y, location of the point, Laplacian value, the size of the feature, direction, value of Hessian, and its descriptor. For particular details on SURF, please refer to Section 3.1.

SURF has been successfully applied to object recognition [103] [55]. The main idea is to apply SURF on the images to locate the touching text from images. Roy et al. [47] has already used SIFT features for extraction of touching text using character templates. Features of character templates are used to localize touching characters. Using only text templates leads to many false positives. In contrast, the system presented in this chapter uses both text and non-text features to localize touching characters, as well as to reduce the number of false positive. Furthermore, the computation of SURF is significantly faster than that of SIFT.

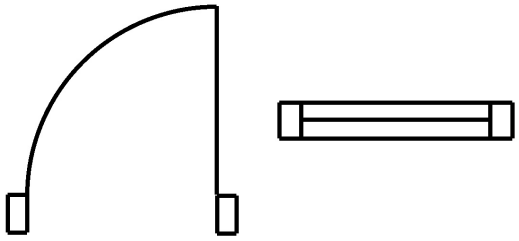


Figure 7.2: Example of non-text component

In the presented approach, first, non-touching text components are extracted from the image. Based on the nature of the image, either the method by Tombre et al. [43] or Ahmed et al. [49] can be used for extraction of non-touching characters. These extracted text components serve as templates for the localization of text components which are touching graphics. If the number of text components extracted by Tombre et al. [43] or Ahmed et al. [49] is very few, then reference templates of typical fonts are also used as templates. To find the font size used in the image, average height(Avg_{height}) and average width(Avg_{width}) are computed from the extracted text components. In addition to alphabet templates, templates for graphical elements are also stored if available, e.g., lines, arcs, and objects (see figure 7.2). This is referred to as graphic template.

In the next step, SURF is applied on every reference text template and all the key points and their respective descriptors are stored as reference text features. Similarly, SURF is applied on the graphics template, and all the key points and their descriptors are stored as reference graphics features. Feature selection is performed to reduce the number of false positives, where all the text descriptors are compared with graphics descriptors. For more details on feature selection, refer to Section 4.2.3. Similar descriptors are removed from both reference text features and reference graphics features.

After removing the similar descriptors from reference features, SURF is applied on the entire graphic image where touching text needs to be localized. This results in a list of key points and their respective descriptors for the graphics image. The descriptors of graphics image (containing touching characters) are compared to the reference text and graphics key points descriptors mentioned above. Finally, the nearest neighbor approach is used to compare these descriptors.

If a key point's nearest neighbor is a text reference key point and the distance between the descriptors is less than $Dist_{text}$, it is marked as a text key point. Similarly, if a

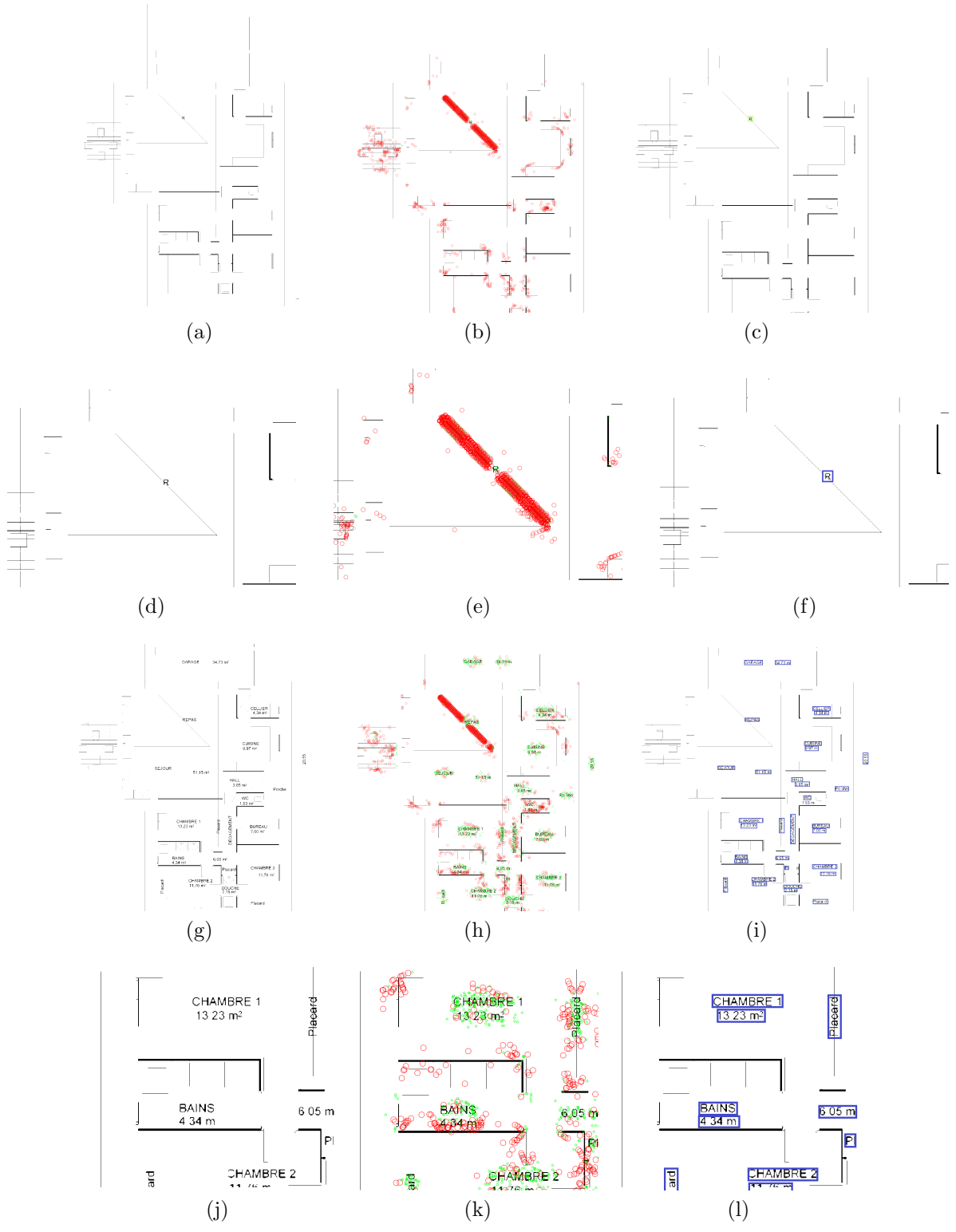


Figure 7.3: Example of localization of text points on floor plan image without external walls. Floor plan without thick walls and touching text (a), extracted text and graphics key points (b), detected text locations (c), (d)(e)(f) are zoomed versions of (a)(b)(c) respectively. Floor plan with non-touching and touching characters without thick walls (g), extracted text and graphics key points (h), detected text locations (I)

key point's nearest neighbor is a graphics reference key point and the distance between descriptors is less than $Dist_{graph}$, it is marked a graphics key point. $Dist_{text}$ and $Dist_{graph}$ are distance thresholds that are computed empirically after investigating the behavior on one reference image. To finally mark a key point as text or graphic, a majority voting is applied based on the neighboring key points. If a key point has more graphic key points as neighbors, it is finally marked as a graphic key point, otherwise it is marked as a text key point.

For extracting the text from the marked text key points, the bounding box of size Avg_{height} and Avg_{width} is constructed on the detected regions, and if this bounding box contains any black component it is marked as touching text.

Figure 7.3a[1] shows the floor plan image where all of the non-touching characters are removed using the method of Ahmed et al. [49]. After applying the nearest neighbor approach, the key points (as illustrated in Figure 7.3b) are extracted. Note that the red circles denote graphics key points and the green circles denote text. Finally, in Fig. 7.3c the resulting bounding boxes are shown. As shown, they only mark the text area which is touching a diagonal line.

To investigate the behavior of the presented method, it is also applied on floor plans where only thick walls were removed and no text extraction is performed. This results in an image, where all of the remaining graphics as well as all text components are present. The results are shown in Figsures 7.3g, 7.3h, and 7.3i, respectively.

7.3 Evaluation

The presented system is evaluated using a data set of original floor plans collected over a period of more than ten years. This data set was primarily introduced for floor plan analysis in Macé [104] and contains 90 floor plan images. Ahmed et al. [49] has performed text/graphics segmentation evaluation on this data set. From the results of the evaluation in Ahmed et al. [49], 327 characters out of 21,737 where those which were difficult to read. Among these 327 characters, 199 characters overlay with graphics.

Analysis of Table 7.1 reveals that the proposed system finds 95% text components which were touching graphics. If these results are combined with the results of text/graphics

[1]Note that in these figures, a zoomed version of an interesting zone in the figure is always shown below (e.g., in Figure 7.3d for Figure 7.3a)

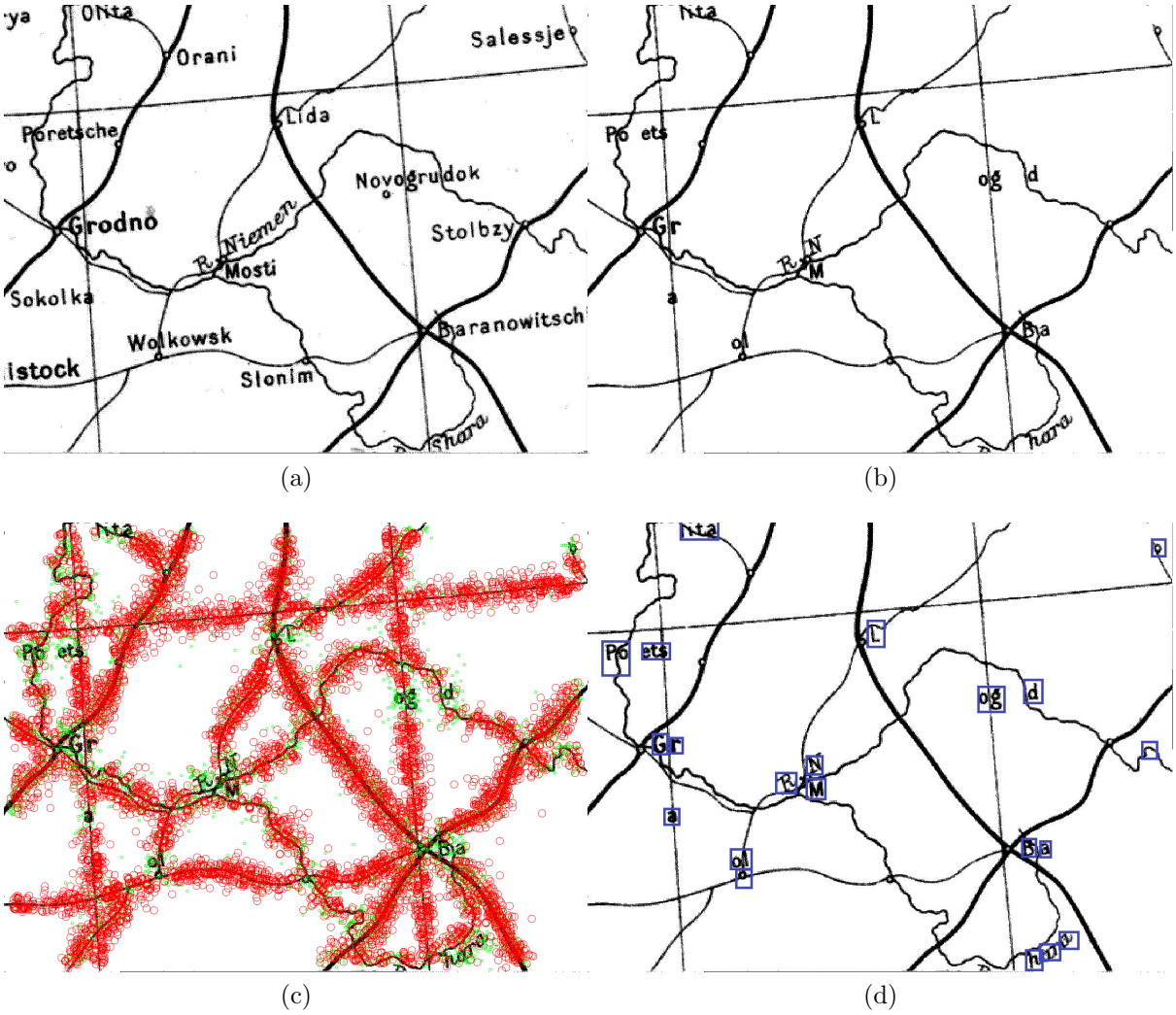


Figure 7.4: Map image (a) After removal of isolated characters (b) text and graphics keypoints marked after comparison

segmentation method in Ahmed et al. [49], the overall recall of text/graphics segmentation method by Ahmed et al. [49] increases significantly. This is because, the touching text components which were missed by the method of Ahmed et al. [49] are successfully extracted by the presented method.

The dataset for maps used by Partha et al. [47] is not publicly available. Therefore, it is not possible to report the results on maps dataset. However, to show that the presented method can be used on map images, the method is tested on a map image present in [47]. Isolated characters are removed using [43]. In Figure 7.4c, it can be seen that all of the touching characters are marked with green key points. It is difficult to judge if the

Touching characters	Number	Percentage (%)
Total	199	100
Retrieved	190	95.48
Missing	9	4.52

Table 7.1: Touching text extraction results

false positives in the map image are errors or not because, on the locations where false positives are detected-there are holes which are very similar to the character "O".

7.4 Conclusion and Future work

This chapter presents a part-based method for extracting text components touching graphics. The method extracts all SURF keypoints of a questioned image and compares them with the keypoints of reference templates from characters and non-characters.

On real floor plan images, it is observed that more than 95 % of the characters were correctly detected. In fact, these characters were actually the problematic characters in the previous text/graphics segmentation method Ahmed et al. [49]. Therefore, the author proposes to use the part-based strategy as a post processing method for text/graphics segmentation methods existing in the literature, e.g., Tombre et al. [43], Ahmed et al. [49], and Fletcher and Kasturi [39]. This method increases the overall recall of the existing methods, as remaining touching characters can be found. Note that it can also be used to increase the precision of existing methods as it can locate graphical elements. This behavior will be investigated on large data sets in the future. Another idea is to use the part-based method as a text/graphics segmentation method alone.

Part II

HYPER-SPECTRAL DOCUMENT ANALYSIS

8

Hyper-spectral Imaging for Signature Analysis

Forensic Handwriting Examiners (FHEs) use Multi-spectral ($4-20$ color channels) and Hyper-spectral (more than 20 color channels) imaging devices to discriminate different inks, same inks-with different aging, alterations to signatures, and for many other related applications. Inspired by the work of FHEs, this chapter presents a novel automatic method for signature segmentation from hyper-spectral document images (240 spectral bands between $400-900$ nm). The presented method is based on the adapted version of generic information segmentation framework presented in Chapter 4 and does not use any structural information, but relies only on the hyper-spectral response of the document regardless of ink color.

8.1 Introduction

Humans are trichromats. According to the trichromatic theory, humans possess three independent visual channels for the perception of colors [107]. In accordance, RGB color space is also defined by the combination of three different colors, i.e., Red, Green, and Blue (RGB) [108]. The reason for aligning the RGB space of cameras, scanners, displays and printers with the trichromatic human vision is that the colors appear real to us. Figure 8.1 shows the color spectrum which ranges from ultraviolet to infrared. The human eye can see objects that lie in the range of the visible spectrum; nothing outside this spectrum is visible to the human eye. In addition, the range from 400-500 nm is

[0]This chapter is an adapted version of the work presented in Ahmed et al. [105] "Hyper-spectral Analysis for Automatic Signature Extraction" In *IGS* 2015, and Ahmed et al. [106] "Automatic Signature Segmentation in hyper-spectral Document Images" In Pattern Recognition Letters (submitted)

perceived by the first cone and corresponds to the blue channel, 450-630 nm is perceived by the second cone and corresponds to the green channel, and 500-700 nm is perceived by the third cone and corresponds to the red channel in the RGB space [108, 109].

Machines, on the other hand, are not trichromats, i.e., they do not only rely on the 3 channel data. In machines, one can have a finer or coarser representation compared to the RGB model, depending on the requirements of the target application. Hyper-spectral imaging (HSI) divides the spectra into a large number of bands [110], which results in a very fine and detailed representation compared to the RGB model. In addition, HSI covers a wider range of spectra, starting from ultraviolet, including visible, and ranging to the infrared region. HSI is already being used in different fields like agriculture [111], medical [112], and mineralogy [113]. In the recent past, a few researchers from document image analysis community have started using HSI for different document analysis tasks. It has been mostly used for historical document analysis [114–120], ink separation [121–123], and document forgery detection [124–126] (See Section 8.2).

This chapter demonstrates an application of HSI to segment a very important biometric modality, i.e., handwritten signatures. While there exist many methods for signature segmentation (see Chapter 5), which use color and/or structural information. In many realistic scenarios, signatures are overlapping with text and it is very difficult to separate the pixels of signature from the text components. Figure 8.2 shows an example of no, partial, and complete overlapping signatures.

Existing methods work with RGB, gray scale, or sometime binary images for signature segmentation. This chapter presents a novel method for automatic signature segmentation from document images. This method uses the power of part-based keypoint detection

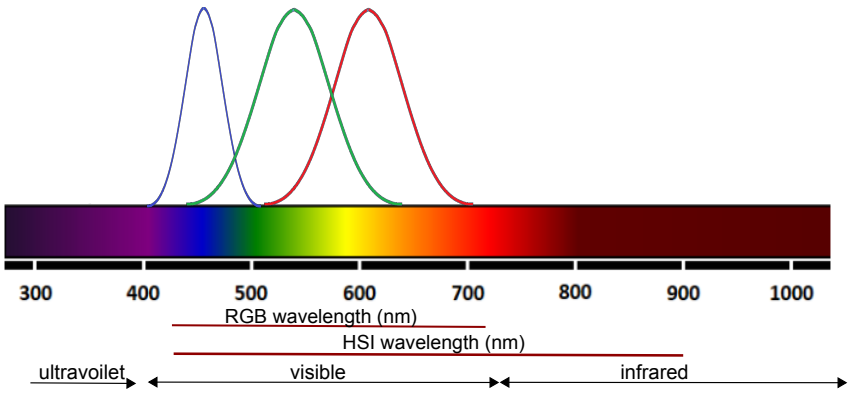


Figure 8.1: The color spectrum.

(a) (b) (c)

Figure 8.2: Signatures occurrences in documents. (a) no overlap, (b) partial overlap, (c) complete overlap

and benefits from the fine and high dimensional representation of hyper-spectral imaging. Existing methods work with RGB, gray scale, or sometime binary images for signature segmentation. This chapter presents a novel method for automatic signature segmentation from document images. This method uses the power of part-based keypoint detection and benefits from the fine and high dimensional representation of hyper-spectral imaging. In particular, SURF is used as part-based keypoint detector. A hyper-spectral camera having a very high spectral resolution, i.e., 2.1 nm, and covering the wavelengths from visible (400 nm) to infrared (900 nm) regions is used in this work. In addition to the signature segmentation system, an HSI documents dataset consisting of 300 documents is also developed and presented here. These documents include non-overlapping as well as overlapping signatures with text and sometimes logos.

The remainder of this chapter is organized as follows. First, Section 8.2 provides an overview of the existing methods that use HSI for document image analysis. Section 8.3 introduces the new HSI dataset presented in this work. Section 8.4 presents the novel automatic signature segmentation method. Finally, Section 8.5 describes the evaluation protocol and experimental results.

8.2 Related Work

This section provides an overview of some of the important works reported in reference to the use of HSI for document image analysis. In the document analysis community, HSI is mostly used for analysis of historical documents, ink mismatch detection, and forgery detection.

For historical documents, HSI has been used primarily for text recovery, character segmentation, and overall document enhancement. P. Shiel et al. [114] used HSI to perform

quality text recovery, segmentation, and dating of historical documents. They performed segmentation on a 16th century paste-down cover and a multi-ink example typical of which is found in the late medieval administrative texts such as Gottingen's kundige book. Aalderink et al. [115] proposed a method for quantitative analysis of historical documents using hyper-spectral imaging. They mapped the distribution of different types of ink and identified the corroded areas within a nineteenth century handwritten letter. In addition, they also proposed a method to enhance the visibility of hidden features like under-drawings on a seventeenth-century historical map. Lettner et al. [116] used spatial (stroke properties) and multi-spectral information in combination with a Markov random field model for character segmentation in ancient documents. D. Goltz et al. [117] used HSI for enhancing the assessment of stains on the surface of historical documents. They use hyper-spectral imaging software (ENVI) for quantitatively assessing the extent of staining in two different documents (a treaty and a prayer book). Hollaus et al. [118] presented a method for enhancement of ancient and degraded writings using Fisher Linear Discriminate Analysis (LDA) on multispectral imaging. Hedjam et al. [119] used HSI for restoration of information from historical documents. The degraded information is restored by extracting, cleaning, and combing spectral response of visible and infrared wavelengths. Recently, Saleem et al. [120] also used LDA on multi-spectral imaging for enhancement of ancient and degraded pieces of handwriting which are barely visible by naked eye. Optical Character Recognition is used to evaluate the enhancement method.

For ink mismatch detection in documents, G. Reed et al. [123] have recently shown that HSI is a useful technique for examination of writing inks. They analyzed the spectral responses of red, blue, and black gel inks and achieved discriminative powers of 1.00, 0.90 and 0.40 for red, blue, and black gel inks respectively. Z. Khan et al. [121] considered handwritten notes drafted in various inks. They presented a publicly available dataset of HSI documents which can be used for ink mismatch detection and applied the k-means clustering for detecting different types of inks in the proposed dataset [122].

Furthermore, HSI is used to perform non-destructive analysis of documents to detect document forgeries. E. B. Brauns et al. [124] proposed a method for non-destructive analysis of potentially fraudulent documents using HSI. They used Fourier transform spectroscopy to achieve spectral discrimination for authentication of written and printed documents. In particular, fuzzy c-means clustering is used to analyze these images. A. Morales et al. [125] proposed a method to detect forgeries in handwritten documents. They used analysis of ink in documents and pen verification using HSI and Least Square

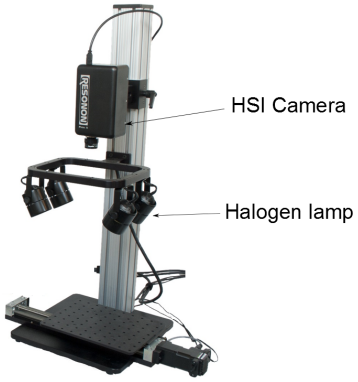


Figure 8.3: HSI scanning setup

SVM classification. The method works for automatic ink type identification, which is tested for 25 different types of pens and achieved an accuracy of 87.5%. C. S. Silva et al. [126] used hyper-spectral imaging near infrared range (HSI-NIR) (from 928 - 2524 nm) to detect forgeries in documents. They analyzed three different types of forgeries i.e., obliterating text, adding text, and crossing lines. Principal component analysis (PCA) along with Multivariate Curve Resolution Alternating Least Squares (MCR-ALS) is used for obliteration and adding text problems. To detect crossing lines MCR-ALS and Partial Least Squares Discriminant Analysis (PLS-DA) were used. The identification rate for obliterating text, adding text, and crossing lines are 43%, 82%, and 85% respectively.

8.3 Dataset

Currently, there are two publicly available datasets that contain information about signature zones, i.e., Tobacco-800 dataset [25] and Maryland Arabic dataset [94]. Both of these datasets contain binarized images, i.e., only one channel data are available for these images. A publicly available hyper-spectral documents dataset is presented by Z. Khan et al. [122]. This dataset contains handwritten text and can be used only for ink separation. There is no publicly available dataset which contains hyper-spectral document images and which can be used for signature segmentation.

This chapter presents a dataset which contains patches from 300 document images, scanned using hyper-spectral camera with a very high spectral resolution of 2.1 nm. The documents used contain printed text mostly in black, but include color graphics and logos. Signatures are performed using different type of pens including oil and gel pens,

having blue and black inks. The dataset of 300 documents is further split into training and test set. The training set contains 30 documents which are representative samples of the complete dataset. The test set includes the remaining 270 documents. Examples of none, partial, and complete overlapping signatures are available in both training and test sets. As ground-truth, bounding boxes of signatures are marked for each document.

The technical details of the camera used for capturing the HSI data are provided in Table 8.1. The setup of HSI system is shown in Figure 8.3. In addition to a high spectral resolution, this camera also covers the complete visible region as well as infrared region until 900 nm. The image scanned using this hyper-spectral camera has 240 bands. This means that each pixel has 240 values in contrast to the 3 values resulting in case of RGB scanning. Figure 8.5 shows an example image from the data.

8.4 Methodology

The presented method for automatic signature segmentation is based an adapted version of information segmentation framework presented in Chapter 4. In addition, it uses the document's spectral response in order to locate the signatures in the document. As mentioned in Section 8.3, the documents are scanned using a hyper-spectral camera having 240 bands ($\lambda_{1...240}$). This results in a representation where each pixel in the document has 240 values.

On inspection of the hyper-spectral response of document images, it was noticed that printer ink has a consistent response across almost all of the 240 band. However, the pen inks had a significant variation in their response across the bands, especially in the infrared wavelengths. This can be observed in Figure 8.4 where spectral responses of page background (white), printed text, and signature pixels are shown. This observation

Parameter	Values
Spectral Range (nm)	400 - 900
Spectral Resolution (nm)	2.1
Spectral Channels	240
Spatial Channels	640
Max Frame Rate (fps)	145
Bit Depth	12

Table 8.1: HSI camera specifications

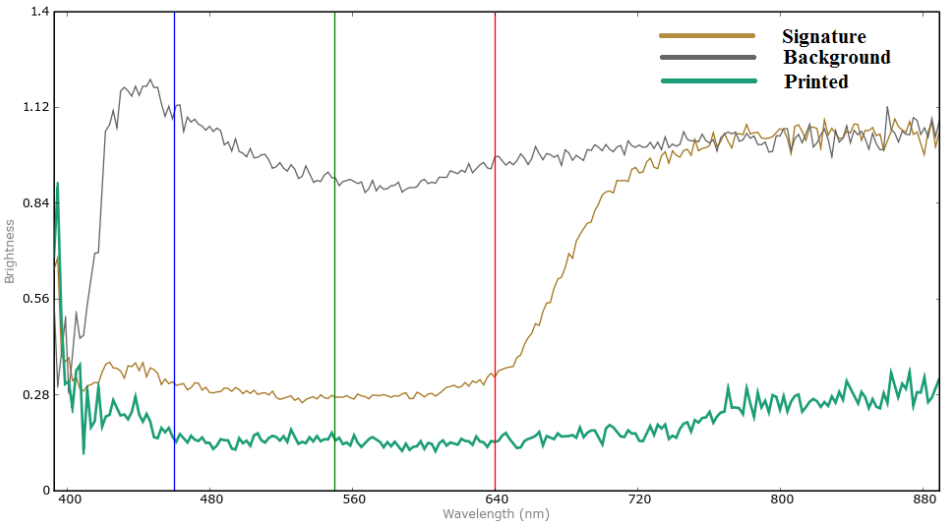


Figure 8.4: Spectral response of page background, machine-printed text, and signature pixels

serves as the building block for our methodology. Based on this observation, the first step is to locate the two most distinguishing bands out of the total 240 bands, i.e.,

1. The band where all of the objects in the document have non-significant response (everything is visible) including signatures. We refer this band as λ_{max}.

2. The band where all of the objects except signatures have non-significant response (everything except signatures is visible). We refer this band as λ_{min}

The $\lambda_{\max}$ and $\lambda_{\min}$ found using a part based keypoint detector. Part based keypoint detector locates most important points in the document, which are referred as keypoints. The Speeded Up Robust Features (SURF) [8] is used for keypoint detection. However, the overall approach is not limited to SURF and can be used with other key-point detection techniques as well, e.g., SIFT [58], FAST [67], or BRISK [2].

For applying the part based keypoint detector, each HSI document image is treated as 240 grayscale images, each containing the spectral response of the document for the corresponding band. Before applying keypoint detector, noise removal is performed to remove small noise sparks which appear on most of the bands. This is done by applying the averaging filter. After noise removal, the SURF keypoint detector is applied on each of the 240 grayscale images generated from the HSI document under consideration. The number of keypoints detected on each image is refereed as δ_n, where n corresponds to the band number. The band with the maximum number of keypoints is refereed as $\lambda_{\max}$ and

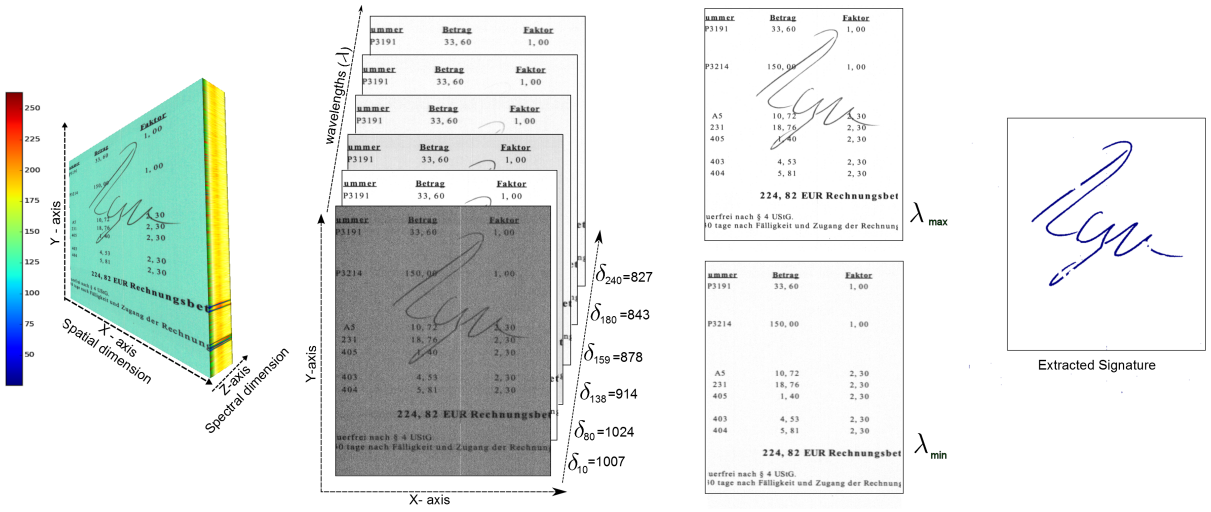


Figure 8.5: Signature segmentation: An example case

the band with the minimum number of keypoints is refereed as λ_{min} (see Equation 8.1). Figure 8.5 shows images corresponding to HSI document and the number of keypoints (δ_n) detected on each image.

$$\lambda_{\text{max / min}}(n) = \begin{cases} \text{max} = \text{n}, & \text{if } \delta_n = \max(\delta_{1\ldots240}). \\ \text{min} = \text{n}, & \text{if } \delta_n = \min(\delta_{1\ldots240}). \end{cases} \tag{8.1}$$

Once these bands ($\lambda_{\text{max}}, \lambda_{\text{min}}$) are located, the next step is to separate the signature pixels from the remaining text/information. To do so, first, morphological opening ($\circ$) is performed on the λ_{max} resulting in noise removal, which is then subtracted from λ_{min}. This results into some noise and signature pixels (λ_{sub}) (see Equation 8.2).

$$\lambda_{\text{sub}} = (\lambda_{\text{max}} \circ \text{mask}_{3\times3}) - \lambda_{\text{min}} \tag{8.2}$$

To remove the noise and get the signature pixels, morphological closing ($\bullet$) is performed on λ_{sub} and the resulting pixels are used as a mask to extract the actual signature pixels from the document. An intersection of signature mask and λ_{max} results in the final signature image (see Equation 8.3). The enclosing rectangle containing the signature mask is refereed as the bounding box of the extracted signature patch. Figure 8.5 shows

the complete workflow and the results produced by the proposed method at each step.

$$\text{signature pixels} = (\lambda_{\text{sub}} \bullet \text{mask}_{3\times3}) \cap \lambda_{\text{max}} \tag{8.3}$$

8.5 Evaluation

The standard precision and recall measures are used to report the performance of the system. Precision represented that how relevant the retrieved bounding boxes were, i.e. what percentage out of the retrieved bounding boxes are corresponding to signatures (see Equation 8.4). Recall indicated that out of all signatures bounding boxes which are present in the document how many are part of the retrieved bounding boxes (see Equation 8.5).

$$\text{Precision} = \frac{(\text{Signature BBox}) \cap (\text{Retrieved BBox})}{(\text{Retrieved BBox})} \tag{8.4}$$

$$\text{Recall} = \frac{(\text{Signature BBox}) \cap (\text{Retrieved BBox})}{(\text{Signature BBox})} \tag{8.5}$$

As mentioned in Section 8.3, only patch level ground truth is available. This means that bounding box corresponding to the signatures is provided in every case. The signature is considered detected if there is at least 50% of overlap between the ground truth and the detected signature patch. Table 8.2 shows the evaluation results of the proposed automatic signature segmentation method.

The proposed method has a recall of 79.31% with the precision of 100%, which means that almost all the signatures are extracted successfully with high precision. Figure 8.6 shows some of the segmentation results. Qualitatively, the correctly segmented signatures are comparable to manually cropped signatures. An important highlight of the proposed method in comparison to the state-of-the-art signature segmentation methods is that, it

Table 8.2: Signature segmentation results on patch level

Metric	Value%
Precision	100
Recall	79.31

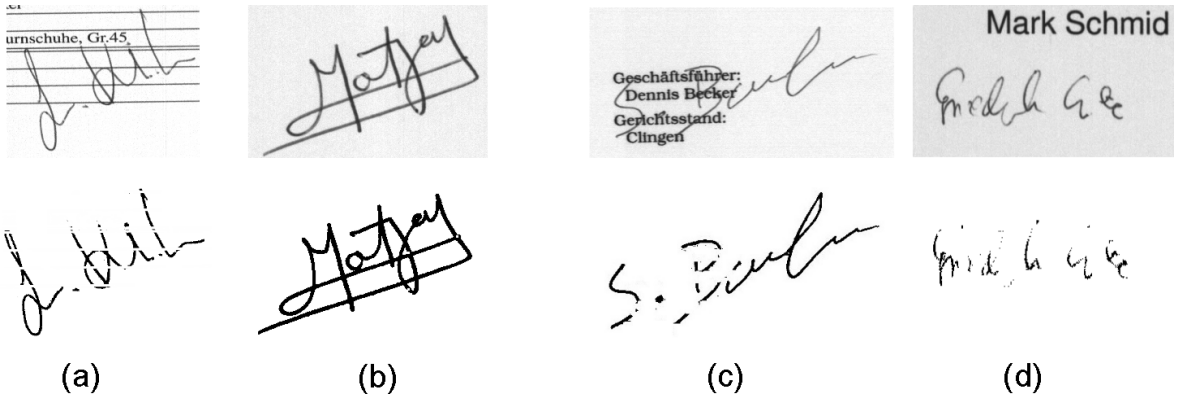


Figure 8.6: Signature extraction results: (a,b,c) Successfully extracted (overlap > 50%) (d) Failure (overlap < 50%)

not only provides the zones of signatures in documents but also extracts the signature pixels. This is very important in cases where signature is overlapping with any other information, because, if only signature zone is returned, it will include extra touching components as well. While for real applications, e.g., writer identification and verification, it is required to have only signature pixels and no other touching component. Figure 8.6 (a,c) shows the cases of overlapping signature and extraction results. It can be seen that the method has extracted all of the pixels which belong to signature only. Figure 8.6 (d) shows the failure case, which occurs due to wrong selection of λ_{min}. This can be improved by analyzing multiple bands around λ_{min} rather than using only a single band i.e., λ_{min}. This is an ongoing research and capturing a large dataset of hyper-spectral images and signatures and making this publicly available is planned for the future.

8.6 Conclusion and Future work

This chapter presents a novel method for automatic signature segmentation from document images. This method is quite unique as it is the first method, ever reported for segmentation of signatures from documents, combining local features of the document with hyperspectral imaging. Further, it is different from all existing signature segmentation methods as it does not use any structural information, but only the hyperspectral response of the document regardless of different colors used in the document. The presented method is also independent of the type, and density of the ink used for writing signatures on documents. A very important aspect of the proposed method is that its performance is consistently very good even when the signatures are overlapping with text

or other information available in the document, e.g., tables, printed text, stamps, logos, and so on. Evaluation results show that proposed method achieved precision and recall of 100% and 79.31% respectively. In the future, it is planned to further increase the scope of the presented method and apply it on diversified documents, like wills, court statements, financial contracts, etc.

Part III

CAMERA-CAPTURED DOCUMENT ANALYSIS

9

A Generic Method for Automatic Ground Truth Generation of Camera-captured Documents

Text recognition is an important part in the analysis of camera-captured documents as there are plenty of services which can be provided based on the recognized text. For example, if text is recognized, one can provide real time translation and information retrieval. Different Optical Character Recognition systems (OCRs) available in the market [129–132] are designed and trained to deal with the distortions and challenges specific to scanned document images.

However, camera-captured document distortions (e.g., blur, perspective distortion, occlusion) are different from that of scanned documents. To enable the current OCRs (developed originally for scanned documents) for camera-captured documents, it is required to train them with data containing distortions available in camera captured documents.

The main problem in building camera based OCRs is the lack of publicly available dataset that can be used for training and testing of character recognition systems for camera-captured documents [133]. One possible solution could be to use different degradation models to build up a large-scale dataset using synthetic data [134, 135]. However, researchers are still of different opinion about either degradation models are true representative of real world data or not. Another possibility could be to generate this dataset by manually extracting words and/or characters from real camera-captured documents and labeling them. However, the manual labeling of each word and/or character in captured

[0]This chapter is an adapted version of the work presented in Ahmed et al. [127] "Automatic Ground Truth Generation of Camera Captured Documents Using Document Image Retrieval" In *ICDAR* 2013, and Ahmed et al. [128] "A Generic Method for Automatic Ground Truth Generation of Camera-captured Documents" In IEEE TPAMI (submitted)

images seem impractical for being very laborious and costly. Hence, there is a strong need of automatic methods capable of generating datasets from real camera-captured text images.

Some methods are available for automatic labeling/ ground truth generation of scanned document images [136–140]. These methods mostly rely on aligning scanned documents with the existing digital versions. However, the existing methods for ground truth generation of scanned documents cannot be applied to camera-captured documents, as they assume that whole document is contained in the scanned image. In addition, these methods are not capable of dealing with problems mostly specific to camera-captured images (blur, perspective distortion, occlusion).

This chapter presents a generic method for automatic labeling/ground-truth generation of camera-captured text document images using a document image retrieval system. The proposed method is automatic and does not require any human intervention in extraction/localization of words and/or characters and their labeling/ground truth generation. A Locally Likely Arrangement Hashing (LLAH) based document retrieval system is used to retrieve and align the electronic version of the document with the captured document image.

In addition to a ground truth generation method, a novel, large, word and character level dataset consisting of one million words and ten million character images extracted from camera-captured text documents is introduced in this chapeter. These images contain real distortions specific to camera-captured images (e.g., blur, perspective distortion, varying lighting). The dataset is generated automatically using the presented automatic labeling method. We refer this dataset as Camera-Captured Characters and Words images (C^3Wi) dataset.

To show the impact of the presented dataset, a Long Short Term Memory (LSTM) based character recognition system that is capable of dealing with the camera based distortion is presented and evaluated. The presented character recognition system is not specific to only camera-captured images but also performs reasonably well on scanned document images by using the same model trained for camera-captured document images.

Furthermore, both commercial as well as open source OCR systems on the C^3Wi) dataset. The aim of this evaluation is to uncover the behavior of these OCRs on real camera-captured document images. The evaluation results show that there is a lot of room for improvements in OCR for camera-captured document images in presence of quality

related distortion (blur, varying lighting conditions, etc.).

The rest of the chapter is organized as follows. Section 9.1 provides an overview of different available datasets and summarizes different approaches for automatic ground truth generation. Section 9.2 presents the automatic ground truth generation method for camera captured document images along with complete evaluation. Section 9.4 provides the description of the presented neural network based character recognizer for camera-captured document images. Section 9.5 provide details on the benchmark tasks performed on the presented dataset. Section 9.6 finally concludes the paper and provides an overview of future research directions.

9.1 Related Work

This section provides an overview of different available datasets and summarizes different approaches for automatic ground truth generation. First, Section 9.1.1 provides an overview of different datasets available for camera-captured documents and natural scene images. Second, Section 9.1.2 provides details about different degradation models for scanned and camera-captured images. In addition, it also provides review of the various existing approaches for automatic ground truth generation.

9.1.1 Existing Datasets

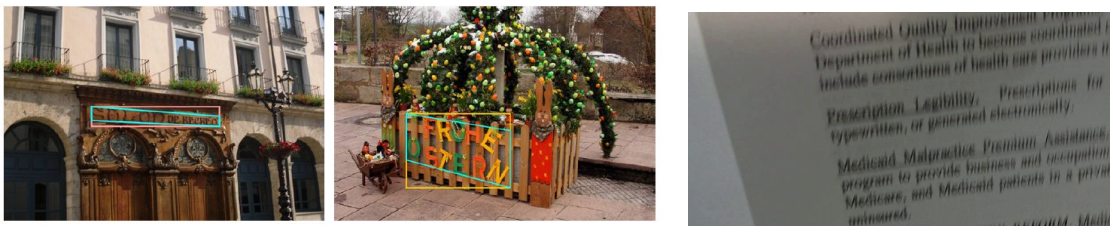
To the best of authors' knowledge, currently there is no 'publicly' available dataset for camera-captured text document images (like books, magazines, article, newspaper) which can be used for training of character recognition systems on camera-captured documents.

Bukhari et al. [141] has introduced a dataset of camera-captured document images. This dataset consist of 100 pages with the text line information. In addition, ground truth text for each page is also provided. It is primarily developed for text line extraction and dewarping. It cannot be used for training of character recognition systems because there is no character, word, or line level text ground truth information available. Kumar et al. [142] has proposed a dataset containing 175 images of 25 documents taken with different camera settings and focus. This dataset can be used for assessing the quality of images, e.g., sharpness score. However, it cannot be used for training of OCRs on camera-captured documents, as there is no character, word, or line level text ground

truth information available. Bissacco et al. [133] has used a dataset of manually labeled documents which were submitted to Google for queries. However, the dataset is not publicly available, and therefore cannot be used for improving other systems.

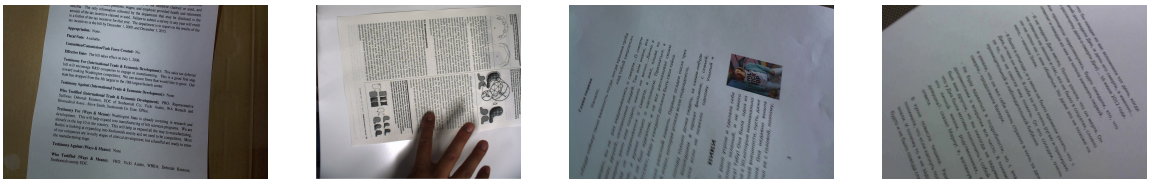
Recently, a camera-captured document OCR competition is organized in ICDAR 2015 with the focus on evaluation of text recognition from images captured by mobile phones [143]. This dataset contains single column 12100 camera-captured document images in English with manually transcribed OCR ground truth (raw text) for complete pages. Similar to Bukhari et al. [141], it cannot be used to train OCRs because there is no character, word, or line level text ground truth information available.

In the last few years, text recognition in natural scene images has gained a lot of attention of researchers. In this context different datasets and systems are developed. The major datasets available are the ones from series of ICDAR Robust Reading Competitions [144–147]. The focus is to enable text recognition in natural scene images, where text is present as either embossed on objects, merged in the background, or is available in arbitrary forms. Figure 9.1 (a) shows natural scene images with text. Similarly, de Campos et al. [148] proposed a dataset consisting up of symbols used in both English and Kannada. It contains characters from natural images, hand drawn characters on tablet PC, and synthesized characters from computer fonts. Netzer et al. [149] introduced a dataset consisting of digits extracted from natural images. The numbers are taken from house numbers in the Google Street View images, and therefore the dataset is known as the Street View House Numbers (SVHN) dataset. However, it only contains digits from natural scene images. Similarly, Nagy et al. [150] proposed a Natural Environment OCR (NEOCR) dataset with a collection of real world images depicting text in different natural variations. Word level ground truth is marked inside the natural images. All of the above-mentioned datasets are developed to deal with text recognition problem



(a) (b)

Figure 9.1: Samples of text in (a) Natural scene image and (b) Camera-captured document image



(a) (b) (c) (d)

Figure 9.2: Samples of camera-captured documents in English (a,b) and Russian (c,d)

in natural images. However, our focus is on documents like books, newspaper, article, magazines, etc., captured using camera, with different camera related distortions e.g., blur, perspective distortion, and occlusion. (Figure 9.1 (a) shows example images from natural scenes with text while Figure 9.2 shows example camera-captured document images). None of the above mentioned datasets contains any samples from the documents similar to those in Figure 9.1 (b) anf Figure 9.2). Therefore, these datasets cannot be used for training of OCRs with the intention to make them working on camera-captured document images.

9.1.2 Ground Truth Generation Methods

One popular method for automatic ground truth generation is to use different image degradation models [151, 152]. An advantage of degradation models is that everything remains electronic, so we do not need to print and scan documents. Degradation models are applied to word or characters to generate images with different possible distortions. Zi [140] used degradation models to synthetic data in different languages, for building datasets, which can be used for training and testing of OCR. Furthermore, some image degradation models have also been proposed for camera-captured documents. Tsuji et al. [134] has proposed a degradation model for low-resolution camera-captured character recognition. The distribution of the degradation parameters is estimated from real images and then applied to build synthetic data. Similarly, Ishida et al. [135] proposed a degradation model of uneven lighting which is used for generative learning. The main problem with degradation models is that they are designed to add limited distortions estimated from distorted image. Thus, it is still debatable that either these models are true representative of real data or not.

In addition to the use of different degradation models, another possibility is to use alignment-based methods where real images are aligned with electronic version to gen-

erate ground truth. Kanungo & Haralick [138, 139] proposed an approach for character level automatic ground truth generation from scanned documents. Documents are created, printed, photocopied, and scanned. Geometric transformation is computed between scanned and ground truth images. Finally, transformation parameters are used to extract the ground truth information for each character. Kim & Kanungo [153] further improved the approach presented by Kanungo & Haralick [138, 139] by using attributed branch-and-bound algorithm for establishing correspondence between the data points of scanned and ground truth images. After establishing the correspondence, ground truth for the scanned image is extracted by transforming the ground truth of the original image.

Similarly, Beusekom et al. [137] proposed automatic ground truth generation for OCR using robust branch and bound search (RAST) [154]. First, global alignment is estimated between the scanned and ground truth images. Then, local alignment is used to adapt the transformation parameters by aligning clusters of nearby connected components. Strecker et al. [136] proposed an approach for ground truth generation of newspaper documents. It is based on synthetic data generated using an automatic layout generation system. The data are printed, degenerated, and scanned. Again, RAST is used to compute the transformation to align the ground truth to the scanned image. The focus of this approach is to create ground truth information for layout analysis.

Note that in the case of scanned documents, complete document image is available, and therefore, transformation between ground truth and scanned image can be computed using alignment techniques mentioned in [136–139]. However, camera-captured documents usually contain mere parts of documents along with other, potentially unnecessary, objects in the background. Figure 9.2 shows some samples of real camera-captured document images. Here, application of the existing ground truth generation methods is not possible due to partial capture and perspective distortions. Note that for camera captured text images, mere scale, translation, and rotation (similarity transformation) is enough which is contrary to scanned text images.

Recently, Chazalon et al. [155] proposed a semi-automatic method for segmentation of camera/mobile captured document image based on color markers detection. Up to the best of authors' knowledge, there is no method available for automatic ground truth generation for camera-captured document images. This makes the contribution of this chapter substantial for document analysis community.

9.2 Automatic Ground Truth Generation: The Presented Approach

The first step in any automatic ground truth generation method is to associate camera-captured images with their electronic versions. In the existing methods for ground truth generation of scanned documents, it is required to manually associate the electronic version of document with the scanned image so that they could be aligned. This manual association limits the efficiency of these methods.

To overcome the manual association and to make the proposed method fully automatic, we used a document image retrieval system. This document image retrieval system automatically retrieves the electronic versions of the camera-captured document images. Therefore, to generate the ground truth, the only thing to do is to capture images of the documents. In the proposed method, an LLAH based document retrieval system is used for retrieving the electronic version of the camera captured text document. This part is referred to as document level matching, Section 9.2.1 provides an overview of this step.

After retrieving the electronic version of a camera-captured document, the next step is to align the camera-captured document with its electronic version. The application of existing alignment methods is not possible on camera-captured documents because of 'partial capture' and 'perspective distortion'. To align a camera-captured document with its electronic version, it is required to perform the following steps:

- **Estimate the regions in electronic version that correspond to camera-captured document.** This estimation is necessary for aligning the parts of electronic version which correspond to a camera-captured document image. It is performed with the help of LLAH, as it not only retrieves the electronic version of captured document, but also provides the estimate of the region/part of electronic version of the document corresponding to the captured document. Section 9.2.1 provides details about LLAH.

- **Alignment of camera-captured document with its corresponding part in electronic document**.

  Using the corresponding region/part estimated by LLAH, part level matching and transformations are performed to align the electronic and the captured image. Section 9.2.2 provides details about part level matching.

- **Words alignment and ground truth extraction**

  Finally, using the parts of image from both the camera-captured and the electronic version of a document, word level matching and transformation is performed to extract corresponding words in both images and their ground truth information from PDF. Section 9.2.3 provides details of word level matching. This step results into word and character images along with their ground truth information.

9.2.1 Document Level Matching

The electronic version of the captured document is required to align a camera-captured document with its electronic version. In the proposed method, we have automated this process by using document level matching. Here, the electronic version of a captured document is automatically retrieved from the database by using an LLAH based document retrieval system. LLAH is used to retrieve the same document from large databases with efficient memory scheme. It has already shown the potential to extract similar documents from the database of 20 million images with retrieval accuracy of more than 99% [156].

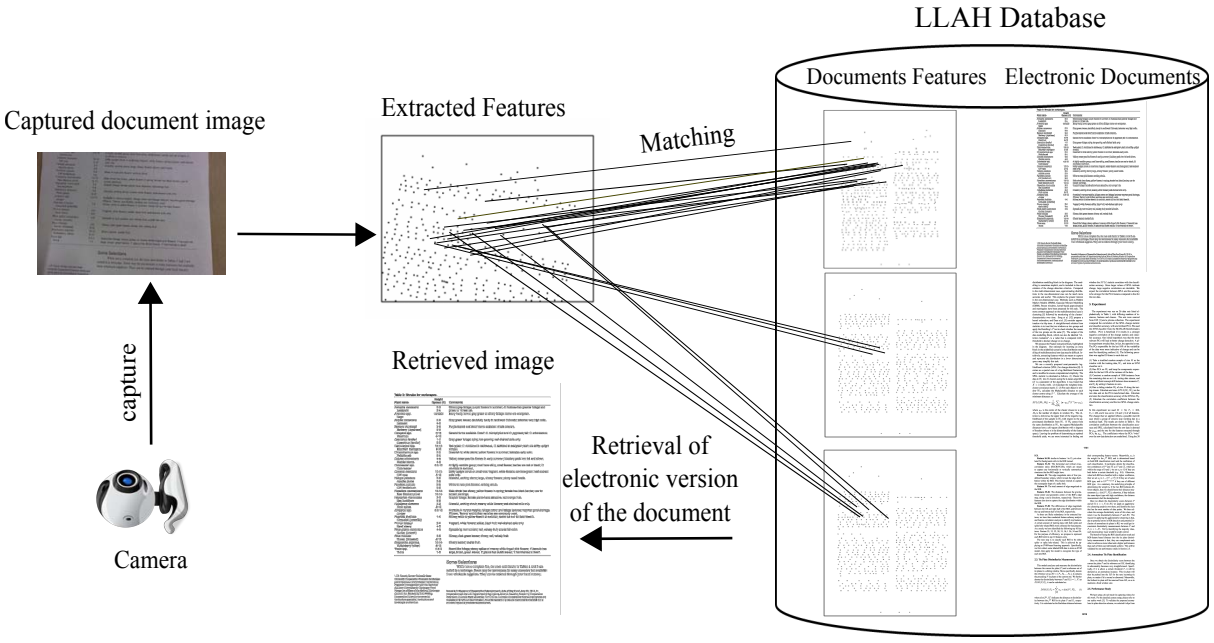


Figure 9.3: Document retrieval with LLAH

Figure 9.3 shows the LLAH based document retrieval system. To use document retrieval system, it is required to first build a database containing electronic version of documents.

To build the database, document images are rendered from their corresponding PDF files at 300 dpi. The documents used to build this database include, proceedings, books, magazines, and other articles.

Here we are summarizing LLAH for completeness; details can be found in [156]. The LLAH extracts local features from camera-captured documents. These features are based on the ratio of the areas of two adjoined triangles made by four coplanar points. First, Gaussian blurring is applied on image which is then converted into feature points (centroid of each connected component). The feature vector is calculated at each feature point by finding its 'n' nearest points. Then 'm' ($m \leq n$) points are chosen from those 'n' points, and among these 'm' points, four are chosen at a time to calculate the affine invariance. This process is repeated for all permutations and 4 from m are chosen. Hence, each feature point will result in $\begin{pmatrix} n \\ m \end{pmatrix}$ descriptors and each descriptor is of $\begin{pmatrix} m \\ 4 \end{pmatrix}$ dimensions. To efficiently match feature vectors, LLAH employs hashing of feature vectors. To obtain hashing index, discretization is performed on the descriptors. Finally, the document ID, point ID, and the discretized feature are stored in a hash table according to the hash index. Hence, each entry in the hash table corresponds to a document with its features.

To retrieve the electronic version of a document from the database, features are extracted from the camera-captured image and compared to features in the database. Electronic version (PDF and image) of the document, having the highest matching score, is returned as the retrieved document.

9.2.2 Part Level Matching

Once electronic version of a camera-captured document is retrieved. The next step is to align the camera-captured document with its electronic version. To do so, it is required to estimate the region of electronic document image (retrieved by document retrieval system) which corresponds to the camera-captured image. This region is computed by making a polygon around the matched points in electronic version of document [156]. Using this corresponding region, the electronic document image is cropped so that only the corresponding part is used for further processing.

To align these regions and to extract ground truth, it is required to first map them into the same space. As compared to scanned documents, camera-captured images contain different types of distortions and transformations (Figure 9.2). Therefore, we need to

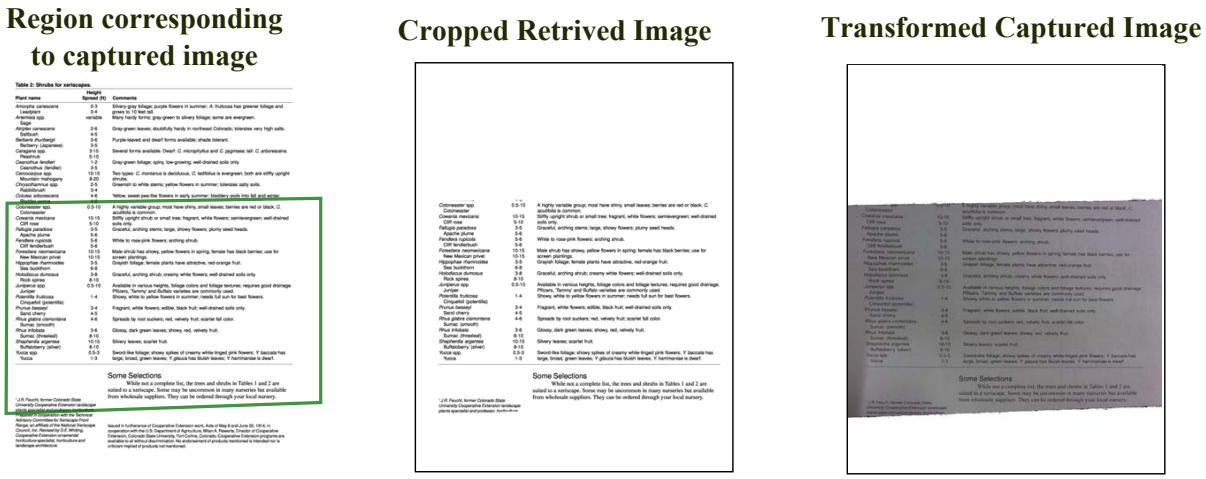


Figure 9.4: Estimation and alignment of document parts

find out transformation parameters which can convert the camera-captured image to the electronic image space. The transformation parameters are computed by using the least square method on the corresponding matched points between the query and the electronic/retrieved version of document image. The computed parameters are further refined with the Levenberg-Marquardt method [157] to reduce the re-projection error. Using these transformation parameters, perspective transformation is applied to the captured image, which maps it to the space of the retrieved document image.

Figure 9.4 shows the cropped electronic document image and the transformed/normalized captured images (captured image after applying perspective transformation) which are further used in word level processing to extract ground truth.

9.2.3 Word Level Matching and Ground Truth Extraction

Figure 9.5 shows the aligned camera-captured and electronic documents. It can be seen that only some parts of both documents (electronic and transformed captured) are perfectly aligned. This is because; the transformation parameters provided by the LLAH are approximated parameters and are not perfect. If these transformation parameters were directly used to extract corresponding ground truth from PDF file, it would lead to false ground truth information for the parts which are not perfectly aligned. The word level matching is performed to avoid this error. Here, the perfectly aligned regions are located so that exactly the same and complete word is cropped from the captured and electronic images.

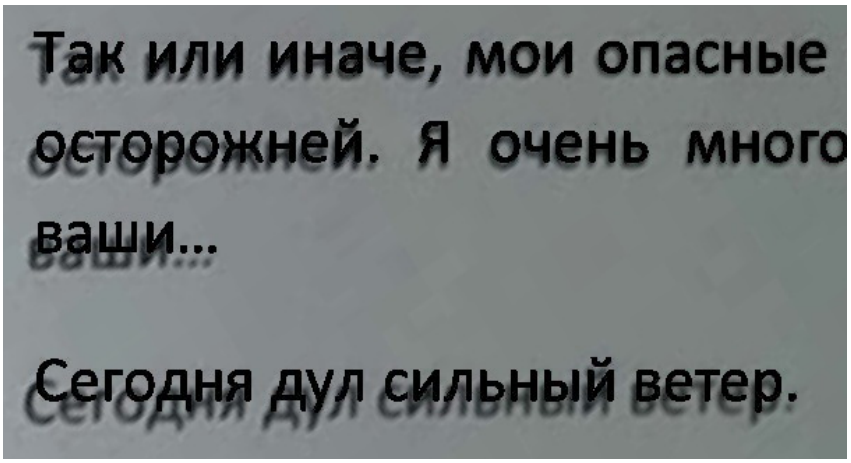


Figure 9.5: Overlapped electronic version and normalized camera-captured images

To find such word regions, the image is converted into word blocks by performing Gaussian smoothing on both the transformed captured image and the cropped electronic image. Bounding boxes are extracted from the smoothed images, where each box corresponds to a word in each image. To find the corresponding words in both images, the distance between their centroids (d_{centroid}) and width (d_{width}) is computed. The distance between centroids and width of bounding boxes is computed using the following equations.

$$d_{\text{centroid}} = \sqrt{(\overline{x}_{\text{capt}} - \overline{x}_{\text{ret}})^2 + (\overline{x}_{\text{capt}} - \overline{y}_{\text{ret}})^2} < \theta_c \tag{9.1}$$

$$d_{\text{width}} = \sqrt{(w_{\text{capt}} - w_{\text{ret}})^2} < \theta_w \tag{9.2}$$

$(\overline{x}_{\text{capt}}, \overline{y}_{\text{capt}}), w_{\text{capt}}$ and $(\overline{x}_{\text{ret}}, \overline{y}_{\text{ret}}), w_{\text{ret}}$ refer to centroids and width of bounding boxes in the normalized/transformed captured and the cropped electronic image. All of the boxes for whom d_{centroid} and d_{width} are less than θ_c and θ_w respectively, are considered as boxes for the same word in both the images. Here, θ_c and θ_w refer to the bounding box distance thresholds for centroid and width, respectively.

We have used $\theta_d = 5$ and $\theta_w = 5$ pixels. This means if two boxes are almost at the same position in both images and their width is also almost the same, then they correspond to the same word in both images. All of the bounding boxes satisfying the criteria of Equations 9.1 and 9.2 are used to crop words from their respective images where no Gaussian smoothing is performed. This results in two images for each word, i.e., the word image from the electronic document image (we call it ground truth image) and the

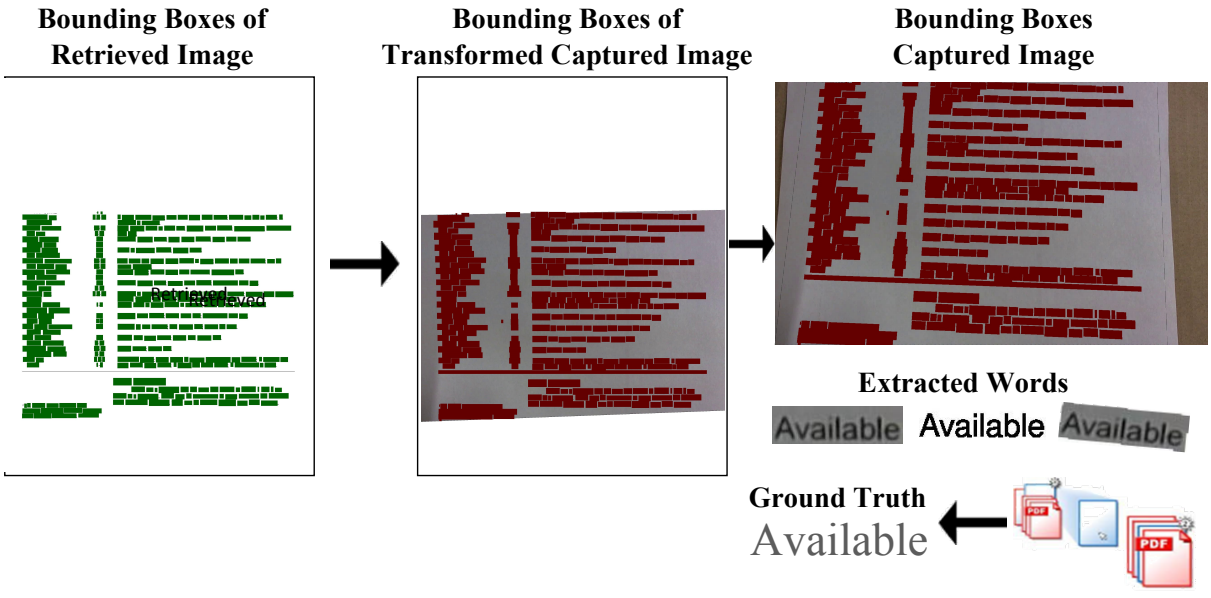


Figure 9.6: Words alignment and ground truth extraction

word image from the transformed captured image.

The word extracted from the transformed/normalized captured image is already normalized in terms of rotation, scale, and skew which were present in the originally captured image. However, the original image with transformations and distortions is of main interest as it can be used for training of systems insensitive to different transformation. To get the original image, inverse transformation is performed on the bounding boxes satisfying criteria set in Equations (1) and (2) in order to map them into the space of the original captured image containing different perspective distortions. The boxes' dimensions after inverse transformation are then used to crop the corresponding words from original captured image. Finally, we have three different images for a word, i.e., from the electronic document image, from the transformed captured image, and the original captured image. Note that the word images extracted from an electronic document are only an add-on, and have nothing to do with the camera-captured document.

Once these images are extracted, the next step is to associate them to their ground truth. To extract the text, we used the bounding box information of the word image from electronic/ground truth image (as this image was rendered from the PDF file) and extract text from the PDF for the bounding box. This extracted text is then saved as text file along with the word images.

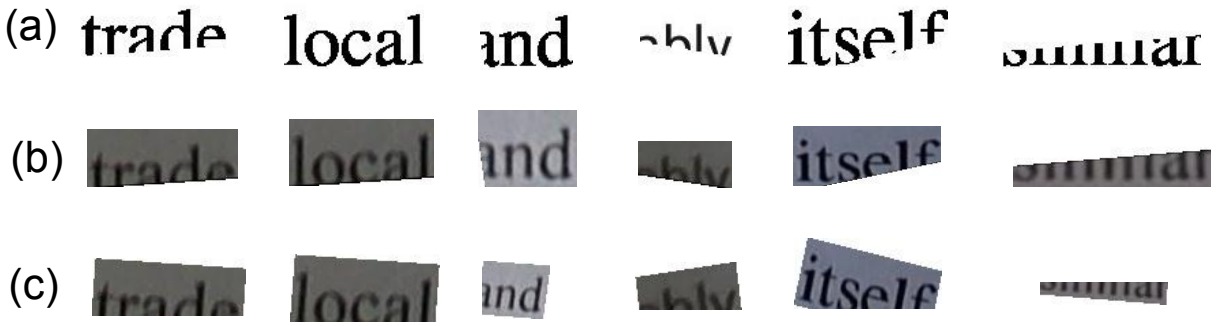


Figure 9.7: Words on border from (a) Retrieved image, (b) Normalized captured image, (c) Captured image

To further extract characters from the word images, character bounding box information is used from PDF file of the retrieved document. In a PDF file, we have information about the bounding box of each character. Using this information, bounding boxes of the character of words satisfying the criteria of Equation 9.1 and 9.2 are extracted. These bounding boxes along with transformation parameters are then used for extracting character images from the original and the normalized/transformed captured images. The text for each character is also saved along with these images.

Finally, we have characters extracted from the captured image and the normalized captured image. Figure 9.9 and 9.10 show the extracted characters and words images.

9.2.4 Special Cases

As mentioned earlier, it is possible that a camera-captured image contain only a part of a document. Therefore, the region of interest could be any irregular polygon. Figure 9.4 shows the estimated irregular polygon in green color. Due to this, the characters and words that occur near or at the border of this region are partially missing. Figure 9.7 shows some example words which occur at border of different camera-captured images. These words, if included directly in the dataset, can cause problems during training, e.g., if a dot of an i is missing then in some fonts it looks like 1 which can increase confusion between different characters. To handle this problem, all the words and characters that occur near border are marked with a flag in their names. This allows separating these words so that they can be handled separately if included in training.

Figure 9.8: Words where human faced difficulty in labeling

9.2.5 Cost Analysis: Human vs. Automatic Method

To get a quantitative measure and to find effectiveness and efficiency of the proposed method, cost analysis between human and the proposed ground truth generation method is performed. Ten documents, captured using camera, were given to a person to perform word and character level labeling. The same documents were given as an input to our system. The person performing labeling task took 670 minutes to label these documents. To crop words from the document it took additional 940 minutes. In total the person took 1610 minutes to extract words and label them. On the other hand, for the same documents, our system was able to extract all words and character images with their ground truth, and normalized images (where they are corrected for different perspective distortion) in less than 2 minutes. This means that the presented automatic method is almost 800 times faster than human. It also confirms the claim that it is not possible to build very large-scale datasets by manual labeling due to extensive cost and time. With the presented approach, it is possible to build large-scale datasets in very short time. The only thing, which needs to do, is document capturing. Rest is managed by the method itself.

Another important benefit of the presented method over human is that the presented method is able to assign ground truth to even severely distorted images where even humans were unable to understand the content. Figure 9.8 shows example words where the human had difficulty in labeling but were successfully labeled by the proposed method.

9.2.6 Evaluation of Automatic Ground Truth Generation Method

To evaluate the precision and prove that the proposed method is generic, two datasets are generated: one in English and other in Russian. The dataset in English consist of one million word and ten million character images. The dataset in Russian contains approx-

imately 100,000 word and 500,000 character images. Documents used for generation of these datasets are diverse and include books, magazine, articles, etc. These documents are captured using different cameras ranging from high-end cameras to normal web-cams.

Manual evaluation is performed to check correctness and quality of the generated ground truth. Out of the generated dataset, 50,000 samples were randomly selected for evaluation. One person has manually inspected all of these samples to find out errors. This manual check shows that more than 99.98% of the extracted samples are correct in term of ground truth as well as the extracted image. A word or character is referred to as correct if and only if the content in cropped word from electronic image, the transformed captured image, the original captured image, and the ground truth text corresponding to these images is the same. While evaluating, it is also taken into account that each image should exactly contain the same information. In addition to camera-captured images, the proposed method is also tested on scanned images, where it has also achieved an accuracy of more than 99.99%. This means that almost all of the images are correctly labeled.

9.3 Camera-Captured Characters and Word images (C^3Wi) Dataset

The dataset is generated using the method proposed in Section 9.2. It contains one million words and ten million character images extracted from different text documents. These characters and words are extracted from diverse collection of documents including conference proceedings, books, magazines, articles, and newspapers. The documents are first captured using three different cameras ranging from normal web cams to high-end cameras, having resolution from two megapixels to eight megapixels. In addition, documents are captured under varying lighting conditions, with different focus, orientation, perspective distortions, and out of focus settings. Figure 9.2 shows sample document captured using different cameras and in different settings. Captured documents are then passed to the automatic ground truth generation method, which extracts word and character images from the camera-captured documents and attach ground truth information from PDF file.

Each word in the dataset has the following three images:

- Ground truth word image: This is a clean word image extracted from the elec-

Figure 9.9: Extracted characters from (a) Normalized captured image, (b) Captured image

Figure 9.10: Word sample from an automatically generated dataset. (a) Ground truth image, (b) Normalized camera-captured image (c) camera-captured image with distortions

tronic version (ground truth) of the camera-captured document. Figure 9.10 (a) shows example ground truth word images extracted by the ground truth generation method.

- Normalized word image: This image is extracted from normalized camera-captured document. This means that it is corrected in terms of perspective distortion, but still contains qualitative distortions like blur, varying lighting condition, etc. Figure 9.10 (b) shows example normalized word images extracted by the ground truth generation method.

- Original camera-captured word image: This image is extracted from the original camera-captured document. It contains various distortions specific to camera-captured images, e.g., perspective distortion, blur, and varying lighting condition. Figure 9.10 (c) shows example camera-captured word images extracted by the presented ground truth generation method.

In addition to these images, a text ground truth is also attached with a word, which contains actual text present in the camera-captured image.

Similarly, each character in the dataset has two images:

- Normalized character image: This image is extracted from normalized camera-captured document. This means that it is corrected in terms of perspective distortion, but still contains qualitative distortions like blur, varying lighting condition, etc. Figure 9.9 (a) shows the example normalized character images extracted by the ground truth generation method.
- Original camera-captured character image: This image is extracted from the original camera-captured document. It contains various distortions specific to camera-captured images, e.g., perspective distortion, blur, and varying lighting condition. Figure 9.9 (b) shows the example camera-captured character images extracted by the ground truth generation method.

For each character image, a ground truth file (containing text) is also associated, which contains characters present in an image.

In total, the dataset contains three million word images along with one million word ground truth text files and twenty million character images with ten million ground truth files.

The Dataset is divided into training, validation, and test set. Training set includes $600,000$ words and six million characters. This means that 60% of the dataset is available for training. The Validation set includes $100,000$ words (one million characters). The test set includes the remaining $300,000$ words and three million character images.

9.4 Neural Network Recognizer: The Presented Character Recognition System

In addition to automatic ground truth generation method and $\mathrm{C^3W}i$ datatset, this chapter also presents a character recognition system for camera-captured document images. The proposed recognition system is based on Long Short Term Memory (LSTM), which is a modified form of Recurrent Neural Network (RNN).

Although RNN performs very well in the sequence classification tasks, it suffers from the vanishing gradient problem. The problem arises when the error signal flowing back-

wards for the weight correction vanish and thus are unable to model long-term dependencies/contextual information. In LSTM, the vanishing gradient problem does not exist and, therefore, LSTM can model contextual information very well.

Another reason for proposing an LSTM based recognizer is that they are able to learn from large unsegmented data and incorporate contextual information. This contextual information is very important in recognition. This means that while recognizing a character it incorporates the information available before the character.

The structure of the LSTM cell can be visualized as in Figure 9.11 and simplified version is mathematically expressed in Eq. 9.3, 9.4 and 9.5. Here, the traditional RNN unit is modified and multiplicative gates, namely input (I), output (O), and forget gates (F), are added. The state of the LSTM cell is preserved internally. The reset operation of the internal memory state is protected with forget gate which determines the reset of memory based on the contextual information.

$$\begin{pmatrix} I \\ F \\ C \\ O \end{pmatrix} = f(W.X^t + W.H_d^{t-1} + W.S_{c,d}^{t-1}) \tag{9.3}$$

$$S_c^t = I.C + S_{c,d}^{t-1}.F_d \tag{9.4}$$

$$H^t = O.f(S_c^t) \tag{9.5}$$

The input, forget, and output gates are denoted by I, F, and O respectively. The t denotes the time-step and in our case, a pixel or a block. The number of recurrent connections are equivalent to the dimensions which are represented by d. It is to be noted that for exploiting the temporal cues for recognition, the word images are scanned in 1D. So, for the equations mentioned above, the value of d is 1.

In offline data it is possible to use both the past and the future contextual information by scanning them from both direction, i.e., left-to-right and right-to-left. An augmentation of the one directional LSTM is the bidirectional long short term memory (BLSTM) [5, 158]. In the proposed method, we used BLSTM where each word image is scanned from left to right and from right to left. This is accomplished by having two one directional LSTM but the scanning is done in different directions. Both of these hidden layers are connected

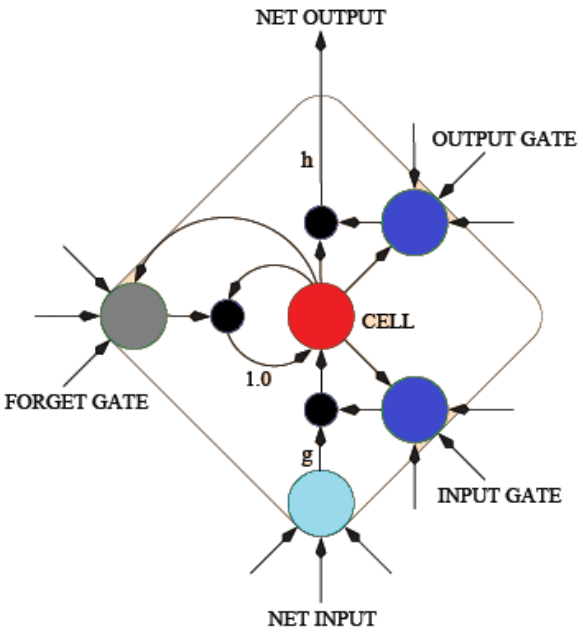


Figure 9.11: LSTM memory block [5]

to output layer for providing the context information from both the past and the future. In this way at a current time step, while predicting a label, we would be able to have the context both from the past and from the future.

Some earlier researchers, like Bissacco et al. [133] used fully connected neural networks. However, segmented characters are required to train their system. Furthermore, to incorporate contextual information they used language modeling. Although in the proposed dataset, we have provided character data as well but we are still using unsegmented data. This is because, with unsegmented data, LSTM is able to automatically learn the context. Furthermore, segmentation of data itself can lead to under and/or over segmentation, which can lead to problems during training, whereas in unsegmented data this problem simply does not exist. RNNs also require pre-segmented data where target has to be specified at each time step for the prediction purpose. This is generally not possible in unsegmented sequential data where the output labels are not aligned with the input sequence. To overcome this difficulty and to process the unsegmented data Connectionist Temporal Classification (CTC) has been used as an output layer of LSTM [159]. The algorithm used in CTC is forward backward algorithm, which requires the labels to be presented in the same order as they appear in the unsegmented input sequence. The combination of LSTM and CTC yielded the state-of-the-art results in handwriting

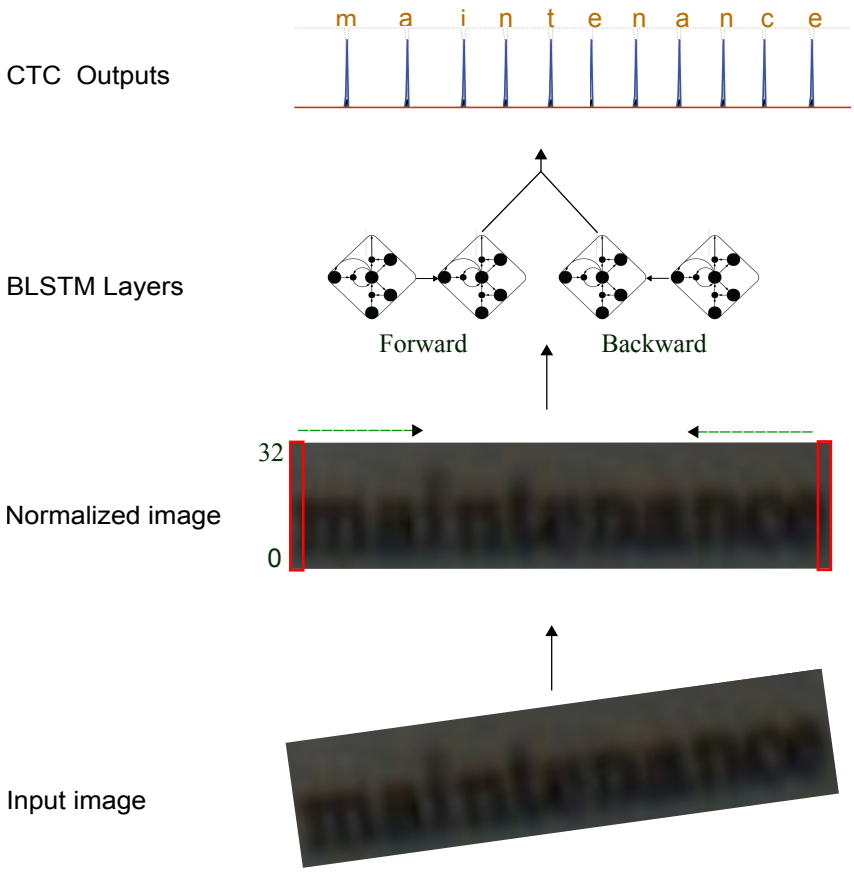


Figure 9.12: Architecture of LSTM based recognizer

analysis [5], printed character recognition [160], and speech recognition [161, 162].

In the proposed system, we used BLSTM architecture with CTC to design the system for recognition of camera-captured document images. BLSTM scans input from both directions and learn by incorporating context into account. Unsegmented word images are given as an input to BLSTM. Contrary to Bissacco et al. [133], where the histogram of oriented gradients (HOG) features are used, the proposed method takes raw pixel values as input for LSTM and no sophisticated features extraction is performed. The motivation behind raw pixels is to avoid handcrafted features and to present LSTM with the complete information so that it can detect and learn relevant features/information automatically. Geometric corrections, e.g., rotation, skew, and slant correction is performed on input images. Furthermore, height normalization is performed on word images that are already corrected in terms of geometric distortions. Each word image is rescaled to the fixed height of 32 pixels. The normalized image is scanned from right to left with a window of size $32X1$ pixels. This scanning results into a sequence that is fed into BLSTM. The

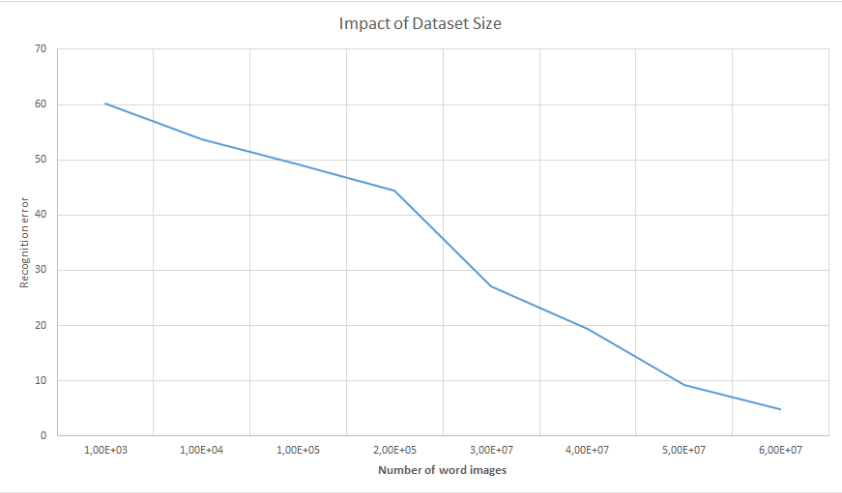


Figure 9.13: Impact of dataset size on recognition error

complete LSTM based recognition system is shown is Figure 9.12. The output of BLSTM hidden layers is fed to the CTC output layer, which produces a probability distribution over character transcriptions.

Note that various sophisticated classifiers, like SVM, cannot be used with large datasets as they can be expensive in time and space. However, LSTM is able to handle and learn from large datasets. Figure 9.13 shows the impact of increase in dataset size on the overall recognition error in the presented system. It can be seen that with the increase in dataset size, overall recognition error drops. The trend in Fig. 9.13 also shows the importance of having large datasets which can be generated using the automatic ground truth generation method presented in this chapter. As this method (explained in Section 9.2) is language independent, we can build very large datasets for different languages, which in turn will result into accurate OCRs for different languages.

9.4.1 Parameter Selection

In LSTM, the hidden layer size is an important parameter. The size of hidden layer is directly proportional to training time. This means that increasing the number of hidden units increases the time required to train the network. In addition to time, hidden layer size also affects the learning of network. A network with few numbers of hidden units results in high recognition error. Whereas, a network with large number of hidden units converges to an over-fitted network. To select an appropriate number of hidden

layers, we trained multiple networks with different hidden units configuration including $40, 60, 80, 100, 120, 140$. We selected network with 100 layers because after 100 error rate on the validation set started increasing.

The training and validation set of C^{3}Wi consisting of $600,000$ and $100,000$ images, respectively, are used to train the network with hidden size of 100, momentum of 0.9, and learning rate of 0.0001.

9.5 Performance Evaluation of the Proposed and Existing OCRs

The aim of this evaluation is to gauge the performance and behavior of existing and proposed character recognition systems on camera-captured document images. To do so, we used the C^{3}Wi dataset, which is generated using the method proposed in Section 9.2. We trained our method on the training set of C^{3}Wi dataset. ABBYY and Tesseract already claim to support camera based OCRs [133,163]. As mentioned in Section 9.3, each word in the dataset has three different images and a text ground truth file, i.e., original camera-captured word image, normalized camera-captured word image, and ground truth image. To have a thorough and in-depth evaluation of OCRs, two different experiments are performed.

- **Experiment 1:** Normalized version of camera-captured word images where original camera-captured images are normalized in terms of perspective distortions (Figure 9.10(b)), are passed to ABBYY, Tesseract, and the proposed LSTM based character recognition system. Note that these images still contain qualitative distortions e.g., blur, and varying lighting.

- **Experiment 2:** Ground truth word image extracted from the electronic version of captured document (Figure 9.10(a)), are passed to ABBYY, Tesseract, and the proposed LSTM based character recognition system.

To find out the accuracy, a Levenshtein distance [164] based accuracy measure is used. This measure includes the number of insertions, deletion, and substitutions, which are necessary for converting a given string into another. Equation 9.6 is used for measuring

the accuracy.

$$Accuracy = \left(1 - \frac{(\text{insertions} + \text{substitutions} + \text{deletions})}{\text{len(ground truth transcription)}}\right) * 100 \tag{9.6}$$

Table 9.1: Recognition accuracy of OCRs for different experiments.

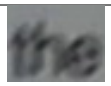
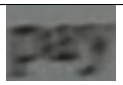
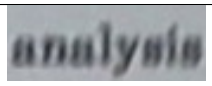
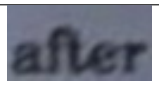
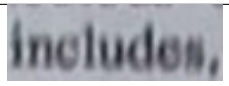
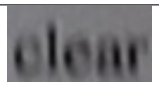
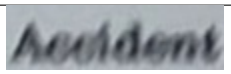
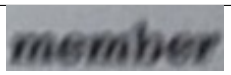
OCR Name	Experiment 1	Experiment 2
Tesseract	50.44	94.77
ABBYY FineReader	75.02	99.41
Proposed System	95.10	97.25

Table 9.1 shows the accuracy for all the experiments. The results of Experiment 1 shows that both Commercial (ABBYY) as well as open source (Tesseract) OCRs fail severely when applied on camera-captured word images. The main reason for this failure is the presence of distortions specific to camera-captured images, e.g., blur, and varying lighting conditions. It is to be noted that images used in this experiment are normalized for geometric distortions like, rotation, skew, slant, and perspective distortion. Even in the absence of perspective distortions, existing OCRs fail. This shows that quality related distortions, e.g., blur and varying lighting have strong impact on recognition in existing OCRs. Table 9.2 shows some sample images along with OCR results of different systems.

To show that OCRs are really working on word images, Experiment 2 was performed. In this experiment, clean word images extracted from electronic versions of camera-captured documents are used. These documents do not contain any geometric or qualitative distortion. These clean ground truth word images are passed to the OCRs. All of the OCRs performed well and achieved very high accuracies. In addition, our proposed system performed even better than the Tesseract and achieved a performance close to ABBYY. It is to be noted that our system was only trained on camera-captured images, and no clean image was used for training. The result of this experiment shows that if trained on degraded images, the proposed system can recognize both degraded as well as clean images. However, other way around is not true, as existing OCRs fail on camera-captured images but perform well on clean word images.

The analysis of the experiments shows that the existing OCRs, which already get very high accuracies on scanned documents, i.e., 99.94%, have a limited performance on camera-captured document images with the best recognition accuracy of 75.02% in case of commercial OCR (ABBYY) and 50.44% in case of open source OCR (Tesseract). On

Table 9.2: Sample results for camera-captured words with distortions

Index	Sample Image	Ground Truth	Tesseract	ABBYY FineReader	Proposed Recognizer
1		to	IO	to	to
2		the		the	thn
3		now	I	no	now
4		pay	’5		py
5		responsibilities	j		responsibiites
6		analysis	umnlym	HNMlyill	annlysis
7		after	.	-fU-r	after
8		act		Ai t	act
9		includes,	Ingludul	Include*,	includes,
10		votes	Will	Virilit	voes
11		clear	vlvur	t IHM	clear
12		Accident	Maiden!		Aceident
13		member	.		meember
14		situation	mum		sltstion
15		generally			genray
16		shall	WNIW	.II	adad
17		Industrial	[Mandi		Industril

deeper examination, it is further observed that main reason for failure of existing OCRs is not the perspective distortion, but the qualitative distortions.

To confirm our findings, we performed another experiment where images with blur and bad lighting were presented to all recognizers. These results are summarized in Table 9.3.

Analysis of results in Table 9.3 confirms that both commercial (ABBY FineReader) as well as open source (Tesseract) OCRs fail severely on images with blur and bad lighting conditions. On the other hand, they are performing well on clean images, regardless of camera-captured or scanned images. The main reason of this failure is qualitative distortions (i.e., blurring and varying lighting conditions), especially, if images are of low contrast, almost all existing OCRs fail to recognize text. While the proposed LSTM based recognizer is able to recognize them with an accuracy of 86.8%.

This effect could be seen in Table 9.2, where the outputs of existing OCRs are not even close to the ground truth. This is because most of these systems are using binarization before recognition. If low contrast images are not binarized properly, there will be too much noise and loss of information, which would result in miss-classification. While the proposed LSTM based recognizer performs reasonably well. It generates outputs close to the ground truth, even for those cases where it is difficult for humans to understand the content, e.g., row 14 and 15 in Table 9.2.

Table 9.3: Recognition accuracy of OCRs on only blur and varying lighting images.

OCR	Accuracy
Tesseract	18.1
ABBYY FineReader	19.57
Proposed System	86.8

Furthermore, note that all the results of the proposed character recognition system are achieved without any language modeling. Analysis of results reveals that there are few mistakes, which can be easily avoided by incorporating language modeling. For example in Table 9.2 the word “voes” can be easily corrected to “votes”.

9.6 Conclusions and Future Work

In this chapter, a novel, generic, method for automatic ground truth generation of camera-captured/scanned document images is presented. The presented method is capable of labeling and generating large-scale datasets very quickly. It is fully automatic and does not require any human intervention for labeling. Evaluation of the sample from generated datasets shows that our system can be successfully applied to generate very large-scale datasets automatically, which is not possible via manual labeling. While comparing the presented ground-truth generation method with humans, it was revealed that the

presented method is able to label even those words where humans face difficulty even in reading, due to bad lighting condition and/or blur in the image. The presented method is generic as it can be used for generation of dataset in different languages (English, Russian, etc.). Furthermore, it is not limited to camera-captured documents and can be applied to scanned images.

In addition to a novel automatic ground truth generation method, a novel dataset of camera-captured documents consisting of one million words and ten million labeled character images is also presented. The presented dataset can be used for training and testing of OCRs for camera-captured documents. Furthermore, along with the dataset, we also presented an LSTM based character recognition system for camera-captured document images. The presented character recognition system is able to learn from large datasets and therefore trained on C^3Wi dataset. Various benchmark tests are performed using the presented C^3Wi dataset to evaluate the performance of different open source (Tesseract [131]), commercial (ABBYY [129, 131]), as well as presented LSTM based character recognition system. Evaluation results show that both commercial (ABBYY with an accuracy of 75.02%) and open source (Tesseract with an accuracy of 75.02%) OCRs fail on camera-captured documents, especially due to qualitative distortions which are quite common in camera-captured documents. Whereas, the presented character recognition system is able to deal with severely blurred and bad lighting images with an overall accuracy of 95.10%.

In the future, it is planned to build dataset for different languages, including Japanese, Arabic, Urdu, and other Indic scripts, as there is already a strong demand for OCR of different languages e.g., Japanese [165], Arabic [166], Indic scipts [167], Urdu [168] and each one needs a different dataset specifically built for that language. Furthermore, it is planned to use the presented C^3Wi) dataset for domain adaptation. This means that training a model on C^3Wi dataset with the aim to make it working on natural scene images.

Part IV

Associated Research

10

Automatic Analysis and Sketch Based Retrieval of Architectural Floor Plans

During design process, architects use existing and already designed buildings as reference. These reference drawings are used to guide solutions for similar architectural situations. Whenever an architect has to solve a new architectural problem, his first task will be to search for similar projects. By studying one or several previous reference projects, the architect tries to derive a solution for his current problem. This is a common approach in architecture, knowing about to use reference projects during the design process and the knowledge of reference projects is an essential skill of an architect.

These days electronic search in architecture is realized by searching for textual annotations. However describing architectural work just using textual information is not sufficient, as verbal descriptions are subjective and often imprecise. Thus a pure textual annotation of floor plans is too fuzzy for an efficient retrieval.

Hence, in order to support an architect in his early design phase, there is a need for a search tool which enables him or her to find similar projects. As already stated, a textual search approach is not sufficient and too fuzzy. Furthermore, as we are dealing with visual information it is more intuitive to formalize a query employing a visual query language. The language should be an abstraction of an original floor plan symbolism, in order to have an intuitive way for architects to formalize a search query.

Langenhan proposes a semantic structure to describe the content of a floor plan based on

[0]This chapter is an adapted version of the work presented in Ahmed et al. [169] "Automatic analysis and sketch-based retrieval of architectural floor plans" In *Pattern Recognition Letters* 2014, and Ahmed et al. [170] "Improved automatic analysis of architectural floor plans" In ICDAR 2013

functional and spatial relations of different structural entities. This formalization results in a graph representation which can be used for the retrieval process. The *a.SCatch* system enables the user to easily access knowledge from past projects. The user searches for semantically similar floor plans just by drawing parts of the new plan.

However, it is a well known problem that it is difficult to generate the semantic database for already existing reference paper floor plans, also known as the bootstrapping problem. Therefore, our second major contribution is an automatic floor plan recognition system which analyzes the floor plans and finally retrieves the corresponding semantic information. Usually, floor plan analysis systems consist of information segmentation, followed by structural analysis and finally, semantic information extraction and alignment. The retrieved structural and semantic information can be saved in a repository for later access during retrieval.

This chapter provides details of the presented architectural floor plan retrieval system. Furthermore, a novel method for analysis and recognition of architectural floor plans is presented in this chapter. Finally, novel post-processing techniques for the semantic floor plan analysis and report on results of the floor plan recognition as well as sketch recognition and floor plan retrieval are presented.

The rest of this chapter is organized as follows. First, Section 10.1 gives an overview of the related work. Subsequently, Section 10.2 introduces the concepts of the a.SCatch system and discusses the details of the floor plan analysis system. Section 10.3 shows the performed experiments. Finally, Section 10.4 concludes the work.

10.1 Related Work

This section provides an overview of related work to the different techniques used in this chapter. Related work can be categorized into architectural background, sketch-based interfaces, symbol spotting, complete floor plan analysis systems, and graph matching.

10.1.1 Architectural Background

Since the middle of the 1990s the approach of applying Case-based reasoning (CBR) to design and architectural tasks has been known as Case-Based Design (CBD). The case-base contains information on buildings that have already been built or designed, enabling

the computer to adapt solutions accordingly, on its own or with help from the architects. Table 10.1 provides a brief overview of some CBD systems based on two studies published by Heylighen and Neuckermans [171] and by Richter et al. [172] regarding the proposed approach. The marked fields show whether the appropriate feature was realized in the concept.

Table 10.1: Overview Case-Based Design systems

	Storage			**Input**			**Output**						
	Floor plans + text	Abstraction	Topology	Graphic	Verbal	Adaption	Reference projects	Applying solutions	Graphical Information	Learning	Subproblems	Semantic net	Analogy
Archie-II	X	X			X		X		X		X	X	
CADRE	X	X	X	X	X	X		X	X	X	X		
FABEL	X	X	X	X	X	X	X	X	X		X		X
IDIOM			X	X	X	X		X	X				
PREC.	X	X			X		X		X		X	X	
SEED L.		X			X	X	X	X	X		X		
SL-CB	X	X			X	X	X	X	X				
TRACE		X	X	X	X		X	X	X				
CaseBook	X		X	X	X		X						
MONEO	X	X			X		X		X				X
CBA	X	X			X		X					X	
DYNAMO	X	X		X	X		X		X	X		X	

The study by Richter et al. [172] identifies an acquisition bottleneck in putting complete case descriptions (problem and solution) into the case-base. We assume this is due to a lack of adequate input strategies, indexing methods and knowledge management procedures. First of all, a user interface should support the graphical sketch-based workflow of architects combined with textual, schematic and tabular input strategies. Secondly, a lightweight indexing strategy is needed in contrast to the overall data storage method used. Thirdly, the problem and solution descriptions need to be stored according to the CBR paradigm. Most of the CBD prototypes do not properly implement this fundamental CBR attribute.

10.1.2 Sketch-Based Interfaces

Sketches are widely used in engineering and architectural fields as they are a familiar, efficient and natural way of expressing certain kinds of ideas. Feng et al. [173], proposed an 2D dynamic programming approach for analyzing hand-drawn electronic circuits. Sezgin

et al. [174] introduced a system that combines multiple sources of knowledge to provide robust early processing for freehand sketching.

Sim-U-Sketch is a sketch-based interface for Simulink[1] [175] where users can construct functional Simulink models simply by drawing sketches on a computer screen. To support iterative design, Sim-U-Sketch allows users to interact with their sketches in real time to modify existing objects and add new ones.

The *COMIC* system [176] is a large European project that studies multi-modal interactions in design applications using pen and speech. In multi-modal system methods, such as mode detection by Willems et al. [177], are sufficient to improve the usability of these systems.

Spatial-Query-by-Sketch proposed by [178] describes a visual spatial query language for geometric information systems. Yaner and Goel [179] examines the retrieval and mapping tasks of visual analogy as constraint satisfaction problems.

10.1.3 Symbol Spotting

A main issue in the floor plan analysis is symbol spotting. Therefore, we list some related work in symbol spotting in this section.

In the past, different pattern recognition techniques have been applied to symbol spotting. Belkasim et al. [180], Li and Shen [181], Adam et al. [182], and Arajo and Kim [183] used feature based description for the purpose of symbol spotting. Similarly symbol spotting based on structural representation of documents has been used by Lladoós et al. [184] and Yan and Wenyin [185]. Furthermore, Tabbone et al. [186] performed symbol spotting based on image segmentation. However, segmentation itself leads to errors, which are then propagated to the recognition.

Another idea to address symbol spotting is to use a vectorial image to spot the symbol rather than using a raster image. Messmer and Bunke [187] and Rusiñol and Lladós [188] used vectorized image for symbol spotting. In addition to vectorization, Rusiñol et al. [189]used indexing techniques to increase the efficiency of symbol spotting techniques. To further increase the scalability of the method Rusiñol et al. [190] used relational indexing of vectorial primitives. Dutta et al. [191] proposed a method using hashing the

[1]Simulink is an environment for multi-domain simulation and Model-Based Design for dynamic and embedded systems. `http://www.mathworks.com/products/simulink/`: Last accessed 04/02/2010

shape descriptors of graph paths.

Nayef and Breuel [192] used geometric primitives as feature points. These feature points are then searched using geometric matching algorithm. To further increase the performance as well as the accuracy of the method, Nayef and Breuel [193] used statistical grouping for segmenting parts from line drawings. After grouping, symbol spotting is performed using the method by Nayef and Breuel [192].

10.1.4 Floor Plan Analysis

In past, floor plan analysis has been performed for different purposes. Aoki et al. [194] and Lladós [195] analyzed hand sketched floor plan and generated respective CAD representation. Whereas, Dosch and Masini [196], Dosch et al. [197], Lu et al. [198], and Or et al. [199] focused on analysis of 2D diagrams of floor plans so that their respective 3D model can be regenerated.

Wessel et al. [200] proposed a method for extracting the room connectivity graphs from 3D architectural models. Based on this connectivity graph a fast and efficient shape retrieval from an architectural database can be achieved. Macé et al. [104] proposed a method to detect rooms in the architectural floor plan images. The method is based on a recursive decomposition of the images until convex regions are found.

Heras et al. [201] performed the segmentation of walls from architectural floor plans. The segmented walls can be used for different purposes like, 3D reconstruction or building boundary construction.

In this chapter, a novel method for automatic analysis of floor plans is introduced. It uses similar ideas as those proposed by Macé et al. [104] and furthermore introduces new ideas like wall edges extraction, boundary detection. This general method can also be used for 3D reconstruction and generating the CAD format. However, the current focus is to extract the structure and the semantic information of the floor plan.

10.1.5 Graph Matching

As for the retrieval algorithm, subgraph-matching approaches seem to be promising. Given the problem that a query graph is formalized at run time and a database of model graphs exists a priori, techniques dealing with these conditions are of interest.

Furthermore, the algorithm has to handle the fact that there might not be an exact match in the database.

Graph matching is challenging in presence of large databases [202, 203]. Consequently, methods for indexing and preprocessing are essential methods. The main idea of the graph filtering is to use simple features to reduce to number of feasible candidates. Another concept clustering is used for grouping similar graphs. In principle, given a similarity (or dissimilarity) measure, such as GED [204], clustering algorithms can be applied.

Messmer and Bunke [205] proposed a decision tree approach for subgraph matching. They are using the permutated adjacency matrix from a graph to build a decision tree. This technique is quite efficient during run time, as a decision tree is generated beforehand which contains all model graphs. Unfortunately, the method has to determine all permutations of the adjacency matrices. Thus, as discussed in their experiments, the method is practically limited to graphs with a maximum 19 vertices.

10.2 Concept of a.SCatch

Based on the results of Langenhan and Petzold [206], the aim of a.SCatch is to implement a system which takes advantage of the semantic information. A semantic search will be realized by sketching a concept of an architectural problem and triggering a search for similar projects of the past. Therefore several subtasks have to be solved:

1. Automatic extraction of the semantic structure of older projects,
2. Extraction of the semantic structure from the sketch of the architect,
3. Retrieval of similar floor plans from the repository,
4. Visualization of the results and the interaction with the user interface.

The following sections will discuss the further details of subtasks of the a.SCatch system. Note that we tried to avoid using many parameters but still few exist. These parameters are optimized on validation set which include 10 images from original floor plans dataset. The test set contains the remaining 80 images, which are used for evaluation of the proposed method.

10.2.1 Automatic Extraction of the Semantic Structure from floor plans

The input data of our system is available in binary format.[2] First, segmentation algorithms are applied to separate the various types of information from one another (see Section 10.2.1). Second, the structure of the extracted information is analyzed to retrieve the structure of the rooms (see Section 10.2.1). Finally, a semantic analysis is applied to retrieve the functions of the rooms, respectively (see Section 10.2.1).

Information Segmentation

Floor plans contain information that collectively help an architect to express the actual dynamics of the building. During floor plan analysis, different types of information need to be interpreted at different points of time. Based on the divide and conquer strategy a process of information segmentation is performed. One of the key points of the proposed method is its fine segmentation of different types of information available in floor plans, e.g., walls, symbols, and text. This is required because information, which is not needed for a specific step, is just noise and might lead to incorrect results.

First, text/graphics segmentation is performed using the methods presented by Ahmed et al. [207]. This method is based on method by Tombre et al. [43] with number of improvements specifically for the floor plans. Text/graphics segmentation separate the text from the graphics in the floor plan image. Text image (see Figure 10.1.2) is later used in semantic analysis for extraction of room functions.

The graphics image is further segmented into walls and other building elements like door, windows, etc. Dosch et al. [197] and Macé et al. [104] used a thick/thin line separation algorithm for the separation of walls from the symbols. This algorithm separates the image into two images, i.e., a *thick lines image* containing the walls (both external and internal walls) and a *thin lines image* containing the symbols. We have enhanced this method by adding a third kind of lines, i.e., *medium lines*. In the proposed method, thick (external walls), medium (internal walls), and thin (symbols) lines image can be achieved by sequentially performing erosions followed by dilations. To get thick lines image, it is performed n times to remove everything but the thick walls, where n is computed

[2]The actual image size is 2479×3508. For making the analysis process more efficient, isotropic down scaling to 1413×2000 has been applied.

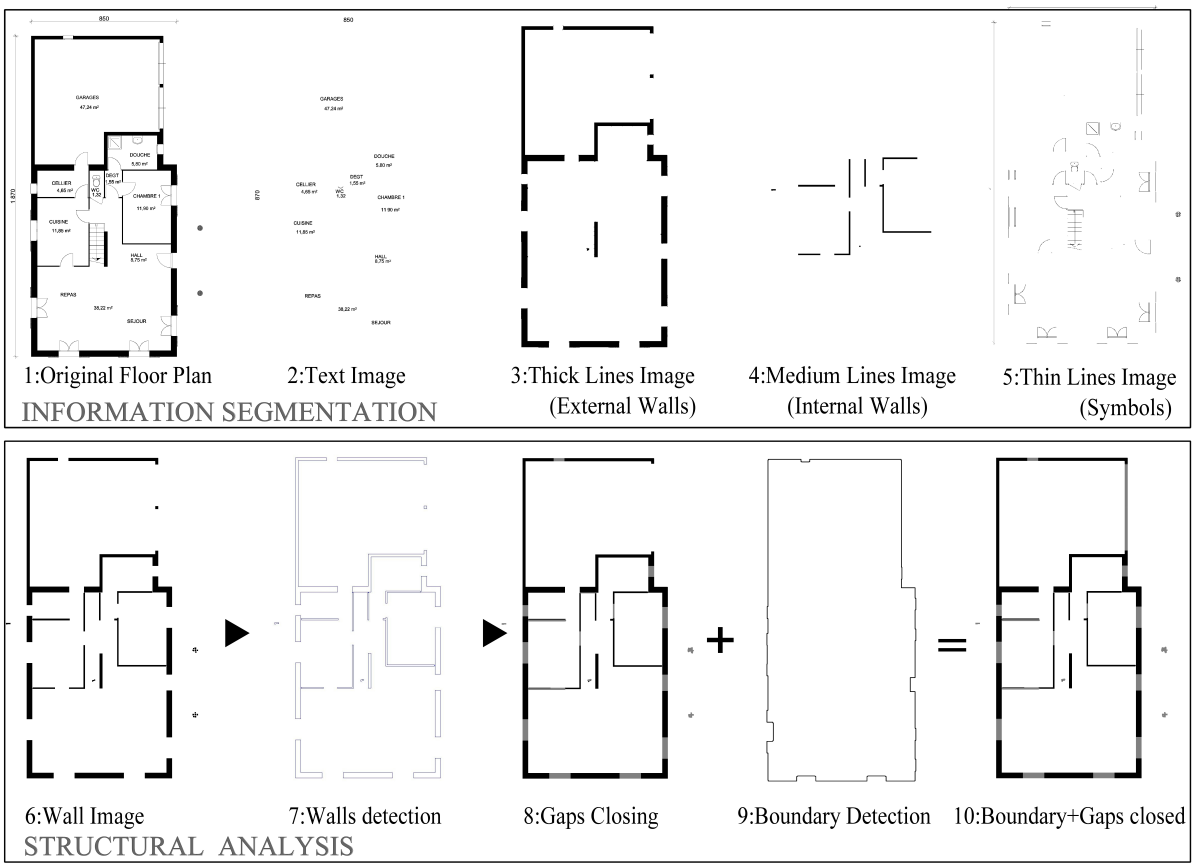


Figure 10.1: Information segmentation and structural analysis

empirically based on thickness of external walls from training set.

The thick line image is then subtracted from graphics image, to obtain an intermediate image, which contains medium and thin lines. On the resulting image, it is performed only one time to get medium lines image. Finally, subtracting thick and medium lines image from graphic image give the thin lines images. Figure 10.1 show the thick, medium, and thin line image extracted from graphics image. This fine segmentation of walls is very useful especially when it is required to save floor plan in CAD format or to do 3D reconstruction of the building. Thick lines can be used to construct the boundary of the building (see section 10.2.1) or 3D reconstruction of the external view of building. Medium lines represent internal structure and partition of the building and can be used to reconstruct the internal structure of the building. Thin lines represent different building elements, e.g., doors, windows, sofas. However, the overall building structure is represented by the walls, therefore thick and medium lines are grouped together to get the walls image (see Fig. 10.1.6).

Structural Analysis

The aim of structural analysis is to extract as much structural information as possible using previously segmented information. Structural analysis begins with the detection of the walls from the wall image as mentioned above. To vectorize the wall image the following approach is used, i.e., first, contours of the wall image are extracted using the method proposed by Suzuki and be [208]. Second, a polygonal approximation is performed on the extracted contours using simple chaining, which results in corner points of each line in contours. Finally, the polygon is constructed for each wall using these corner points. Each polygon represents a wall in the wall image.

The wall edges are then extracted from the detected walls to close the gaps between the walls. These gaps occur at elements like doors, windows, or sometime at gates. The process extracts all edges where those elements are likely to be found. To extract these edges we introduced convex/concave hypothesis. According to this hypothesis, a line of polygon is selected as wall edge if it is short and is either convex or concave (see Fig. 10.2a).

$$Walledge = \begin{cases} Convex & both\ angles\ are > 90^\circ \\ Concave & both\ angles\ are < 90^\circ \end{cases}$$

Figure 10.1.7 shows the extracted wall edges from the walls image according to the convex/concave hypothesis. As a next step the gaps between the extracted edges are closed. Note, that here focus is to close only those gaps where windows or doors are likely to be found based on empirically defined thresholds T_{merge}. T_{merge} is selected based on the size of doors and windows which are inside the building and is optimized on validation set. However, gaps at the outer walls are often larger than gaps occurring at doors inside the building. In order to merge even those larger gaps we compute the outer wall image and use boundary image of the building.

Extraction of the building boundary is another key point of the proposed approach. The extracted building boundary is used to close the large gaps, which are normally due to gates and can be used to get the external structure of a building.

To extract the building boundary a convex hull of the wall image is created, and the portion of the floor plan image, which is inside the convex hull is extracted, neglecting everything outside. Horizontal and vertical smearing is performed on extracted image to fill the gaps between the lines corresponding to windows and gates. To remove all the

lines that are not part of the building structure (often they correspond to measurements), erosion and dilation is performed on the smeared image. After removal of these lines, we can directly extract the external contours of the image. These contours approximate the building boundary, i.e., the external walls. In our experiments described in Section 10.3.1 we show the influence of this particular processing step.

Semantic Analysis

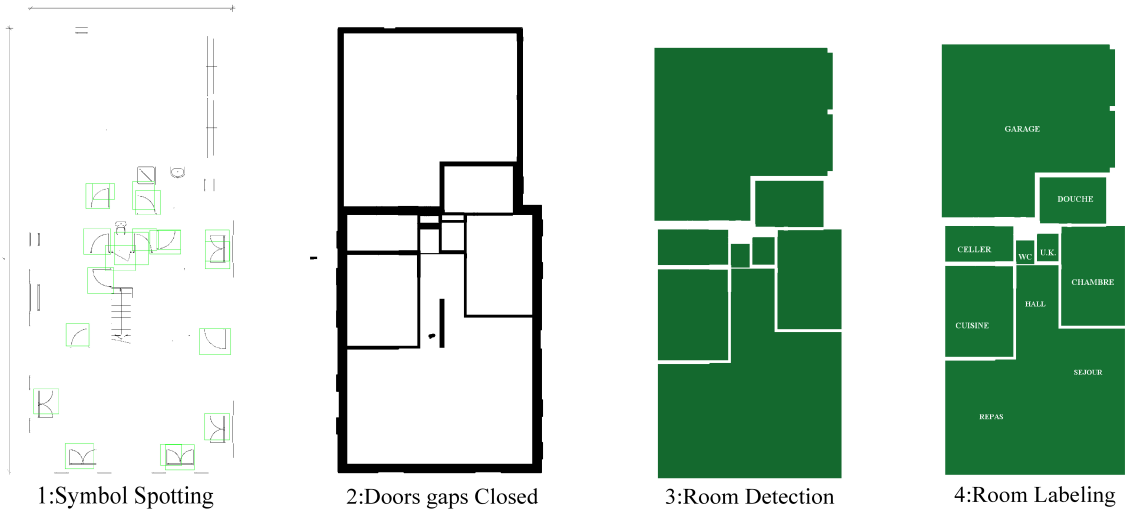
The aim of semantic analysis is to extract the semantic information of the floor plan. While it is easy for a human to gather this information, its automation involves a high complexity. Semantic analysis spots different building elements in the floor plan and interprets them with respect to their context. In this chapter we use the speeded up robust features (SURF) [103], which is a robust, translation, rotation, and scale invariant representation method. It first extracts the key points/points of interest from the image. Then each key point is represented by a discriminative descriptor. A standard door image serves as a reference template for SURF. Mainly arc is detected by SURF, therefore it able to detect both left and right doors.

Note that some erroneous symbols have been extracted by our approach. At a later step these symbol positions are matched with the gaps found during wall edge detection. Only those results, which overlap with gates, are taken into account as actual doors. Figure 10.2b.2 shows the image where the gaps at the doors are closed.

To detect the actual bounds of rooms, the image with the closed gaps is inverted and connected component analysis is performed on it. All of the very small connected components are removed, whereas each of the remaining connected component is referred as room. The detected rooms can be found in Figure 10.2b.3. After detection of rooms the next step is to define there functions like WC, Living room etc. In order to find the function of each room, the text layer from the information segmentation as well as the connected component of the room is used. In particular, all text components, which lie in the boundary of a room, are taken into account. After extraction of the room text, horizontal and vertical smearing is performed on the extracted text to merge the neighboring characters, resulting in the bounds for words. Using the bounding boxes all the words are rotated to a horizontal direction and OCR[3] is performed on them. The OCR result is then compared to rooms title dictionary and the closest title according to the

[3]Tessarect has been used for performing OCR.

(a) A convex, concave, and non edge



(b) Semantic Analysis

Figure 10.2: Convex-Concave hypothesis, and semantic analysis

Levenshtein distance is assigned to the room. Note that before applying the dictionary, all the digits and special characters are removed from the OCR result. The rooms, which do not have any physical partition, contain more than one label.

10.2.2 Sketch-based Retrieval

As we are dealing with visual information, and an exclusive textual description of floor plans is too fuzzy, we propose a simplified visual query language. Whenever the architect is searching the repository, he formalizes his query as a sketch, similar to the fundamental concepts of Spatial-Query-by-Sketch proposed by Egenhofer [178]. Initially, the architect sketches a floor plan with the associated rooms, zones and units (see Fig. 10.6b). The respective online data of the pen device used to detect the geometrical shapes (see Fig. 10.6c) representing the concepts and gestures, which indicate the connection type.

Sketch recognition is performed using the following procedure. During sketching the online pen data is cached. As soon as the user stops drawing for a certain period of time, the recognition of the previous components starts. For the shape detection we used the

Vision Objects shape detection[4].

Currently we use the following visual query language:

- **Rectangles** represent **structural entities**,
- **Enclosings** imply **part-of relation**,
- **Single lines** indicate **adjacent connections**,
- **Two parallel lines** indicate **direct connections**.

The schematic abstraction is now interpreted and translated into the proposed semantic structure by Langenhan and Petzold [206]. Finally, when the user triggers the search, the scene, which is composed of rectangles and lines, will be analyzed. The type of the drawn entity or connection currently is defined by the user by using handwritten annotations on the rectangles. Further work is to interpret the natural symbolism of architects instead of the simplified query language.

10.2.3 Graph Structure

In this section, we discuss how the extracted graph structure can be used for the retrieval. We propose our concept for the retrieval. The extracted semantics are represented as a graph $G = (V, E)$. The vertices V have a type T_{vertex} reflecting a level, unit, zone or room. The edges E also have different types T_{edge} indicating if the vertices are connected directly or are just adjacent, both of these relations are symmetric. The $part\_of$ relation indicates which vertex adheres to a vertex of a superior type T_{vertex}, for instance a sleeping room which is part of a sleeping zone.

The retrieval algorithm is based on a modified version of Messmer and Bunke [205] algorithm, where the row column vectors of the adjacency matrices are arranged in a decision tree. The modified method Weber et al. [209] introduces a well-founded total order on the graph labels. In the original method Messmer and Bunke [205] all permutations of the adjacency matrix, representing the graph, have to be determined. Whereas for the proposed method the permutations, which violate the introduced order, are not allowed. Hence the number of possible permutations is limited. The row-column vectors for each permutation of the adjacency matrix are used to compile the decision tree. The decision tree contains multiple graphs and acts as an index for real-time sub-graph matching.

[4] `http://www.visionobjects.com/`: Last accessed 22/09/2015

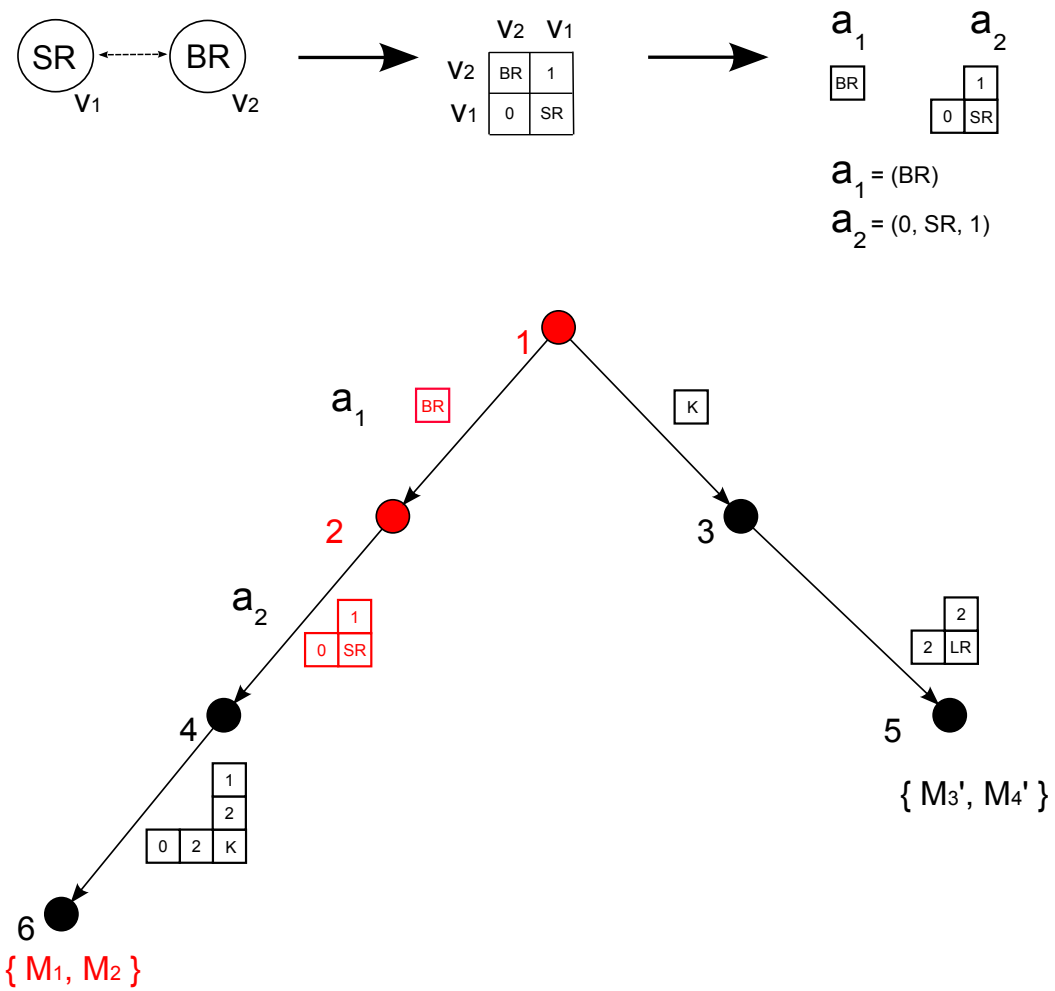


Figure 10.3: The query graph with its two vertices v_1 and v_2 is represented using an adjacency matrix. The ordering of the vertices has to fulfill the well-founded order. The matrix is the split into its row column vectors a_1, a_2. These vectors will be used to find the path in the decision tree, which is used for retrieval.

So, during run-time the decision tree is loaded into memory and by traversing the decision tree, the corresponding subgraph matrices are classified. For the query graph Q the adjacency matrix M is determined following the constraints defined by ordering. Afterwards the adjacency matrix is split up into row-column vectors a_i. For each level i of the decision tree the corresponding row-column vector a_i is used to find the next node in the decision tree using an index structure. Figure 10.3 provides an illustration of a sample query, where a sleeping room (SR) should be directly connected to a bath room (BR). Further details are presented in Weber et al. [209] as well as experiments on graphs with up to 30 vertices.

Our current work focuses on researching approximate approaches for solving the subgraph-matching problem. Furthermore, we are researching techniques to cluster the graph in our repository, by comparing floor plan graphs among each other and group similar ones together. Thus, if a query is not similar to a graph of a cluster it might also be not similar to the other graphs in the cluster.

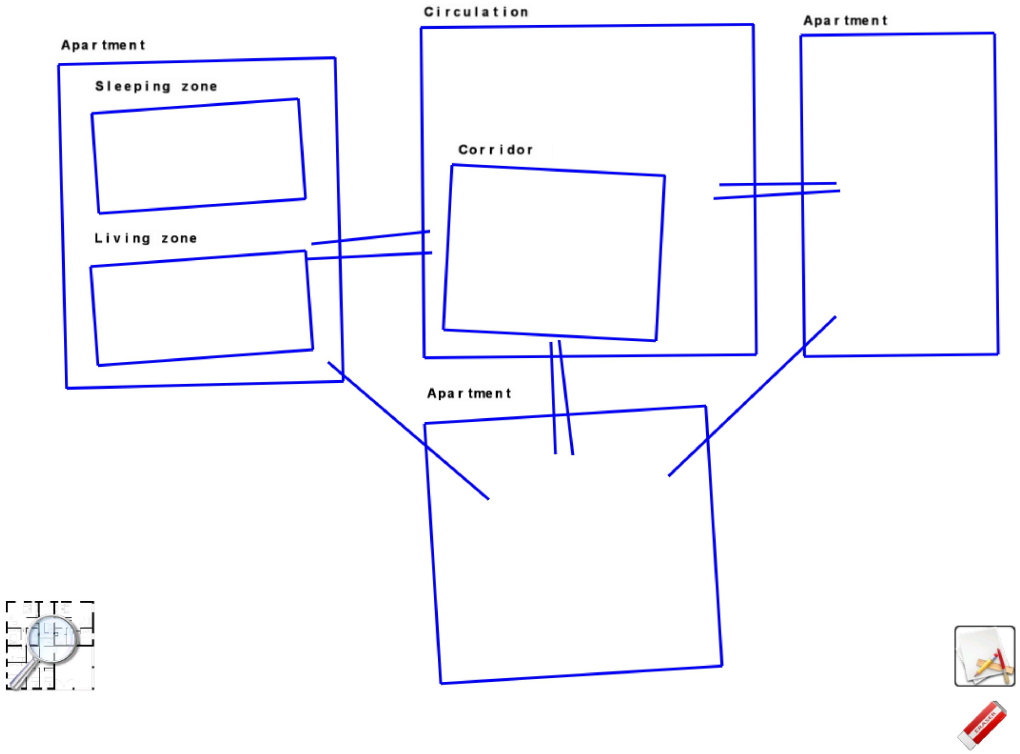
10.2.4 User Interface

As the pen and touch paradigm is more intuitive for the work of an architect, the prototype is implemented for the *Touch & Write* table. The *Touch & Write* table [210] combines the paradigm of multi-touch input and pen input. Architects prefer to sketch in their initial design phase. A pen gives them more freedom than using a mouse with Computer Aided Design (CAD) software. Using the *Touch & Write* pen device to draw in a digital environment allows more immediate interaction and the architects immediately benefit from the digital representation of their drawings.

The pen is an adequate tool to sketch the current architectural problem. The results of the semantic search will be represented as graphical information and the touch interaction is an intuitive metaphor for interacting with the displayed information. For example, the architect is able interact with the graphical information using simple and intuitive gestures to zoom or navigate within the floor plan.

The a.SCatch system provides tools for two purposes. First, an input interface offers the architect a possibility to edit and correct the results of the automatic room and interconnection detection discussed in Section 10.2.1. Here the pen device is used to frame rooms, zones or units in a floor plan.

Second, the architect can sketch a query by using the discussed visual query language. The results are displayed as images ordered according to the calculated similarity measure. The interaction with the result is done by using touch gestures (see Figure 10.4).



(a) a.SCatch retrieval interface.

Figure 10.4: a.SCatch retrieval interface.

10.3 Experiments

10.3.1 Floorplan Analysis Evaluation

Our system is evaluated using a data set containing original floor plan images. This data set was introduced by Macé et al. [104] and contains the floor plan images from the period of more the ten years. The size of each floor plan image in the data set is 2479×3508. All floor plans are binarized to ensure that only structural information of the floor plans is used for the analysis (and not the color information).

In order to report the accuracy of our system, we use the protocol introduced by Phillips and Chhabra [211]. It allows reporting *exact match* (one to one) as well as *partial matches* (one to many and many to one). For further details refer to [211].

Table 10.2 shows the results of rooms detection over the floor plan images dataset. The overall detection rate is 94.88 %, which is approximately 10 % higher than the 85 % achieved in the reference system by Macé et al. [104]. More remarkably, the recogni-

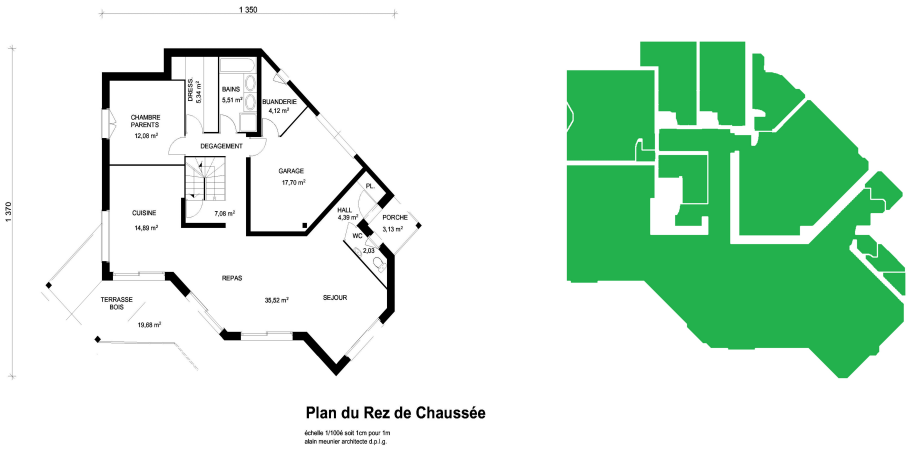


Figure 10.5: Room detection result

Table 10.2: Room detection results

	Macé et al. [104]	Proposed	No boundary det.
Detection rate (%)	85	94.88	77.47
Rec. accuracy (%)	69	81.3	86.88
One to many count	2	1.48	1.01
Many to one count	0.76	2.14	1.58

tion accuracy has been improved by 12.30 %. For around 20 % of the images we received the recognition accuracy and detection rate both greater than 90 %. In the worst case, the recognition accuracy and detection rate of our system were still 50 % and 66.66 % respectively. A further analysis shows the influence of the boundary detection (last column in Tab. 10.2) which was introduced in this chapter. The detection rate is significantly improved.

The analysis of results in Table 10.2 reveals that our system has a good recognition accuracy and detection rate, along with less one to many count on average. This is because, a region is split into sub-regions only when a door or physical partition is found, in contrast to Macé et al. [104] where regions are divided based on polygonal partitioning. To further reduce over segmentation, gap closing on the location where doors are detected need to be improved.

If there is no physical partition in the detected room no division is performed, which results in reduced segmentation overhead. Figure 10.5 shows a very difficult example

Table 10.3: Complexity and detection rate for each query

Query	**Quadrangles**		**Adjacent C.**		**Direct C.**	
1	**#**	**Corr**	**#**	**Corr**	**#**	**Corr**
1	8	0.96	2	0.95	2	0.8
2	5	0.96	2	09	3	0.9
3	8	0.93	2	0.95	5	0.7
4	6	0.95	3	0.93	1	0.7
5	3	1.0	2	1.0	1	1.0
6	9	0.99	3	0.97	5	0.96
7	6	0.98	2	0.95	4	0.65
8	9	0.99	4	0.93	4	0.9
9	10	0.97	1	0.8	4	0.9
10	3	0.97	1	0.9	2	0.9
Overall		0.97		0.93		0.86

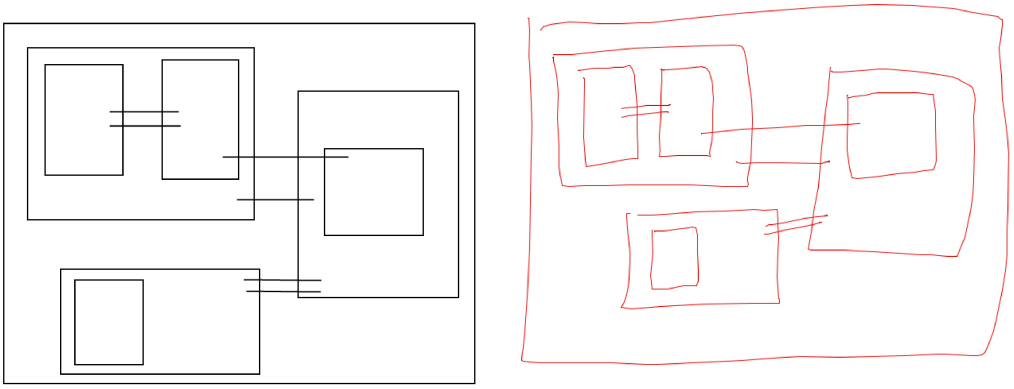
where large regions which do not contain any physical partition are marked as a single room. To further split these regions detailed semantic analysis can be done, in order to split large regions based on the measurement and text information available in the floor plan. This trend can also be seen in Table 10.2, where the *many to one count* is higher.

10.3.2 Retrieval Evaluation

The query generation is a crucial part of the a.SCatch system, thus the first experiments focus on the visual language. We defined ten example queries covering different complexity levels (see Table 10.3) and asked ten participants to copy these sketches, resulting in a total of 100 sketches. The participants were male and female students in the age between 23 and 29 years. All sketches were drawn on the *Touch & Write* and the handwritten strokes were recorded. To assess the pure recognition performance, we did not give a direct feedback of the recognized shapes. In order to measure the accuracy of the detection algorithm we count the correctly detected quadrangles and connections.

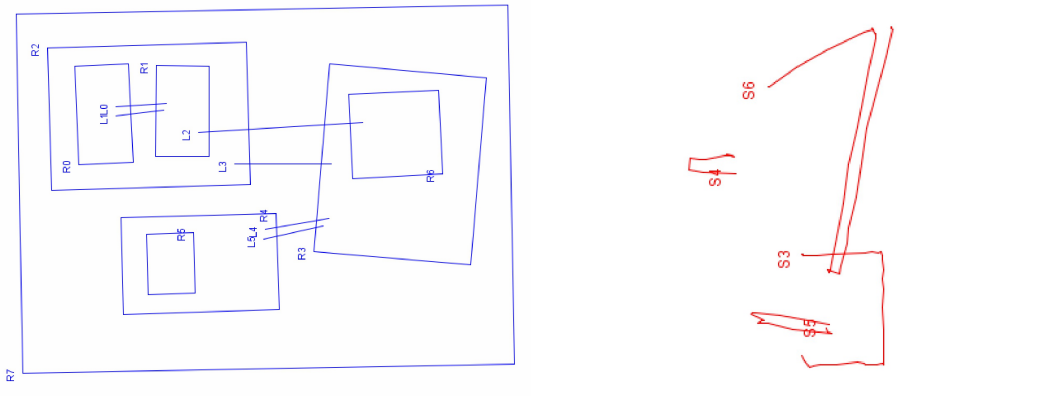
An example of a query taken for the evaluation is given in Fig. 10.6a. Furthermore Fig. 10.6b shows the recorded sketch of a participant. The detected shapes of the shape detection algorithm are illustrated in Fig. 10.6c.

For the evaluation, we distinguished between detection rates for quadrangles, adjacent and direct connections. Table 10.3 shows the detection rate for each query. As can be seen, the detection rates for the quadrangles are very promising. Most of the other errors



(a) Query 3

(b) Sketch of participant 2



(c) Recognized shapes.

(d) Unrecognized strokes

Figure 10.6: Visualization of the results

are due to missing quadrangles. Note that whenever a quadrangle is not detected all corresponding connections with this quadrangle will also be wrongly detected. Considering Query 2, 5, and 10, one can see that a high recognition rate of the quadrangles corresponds to a high recognition rate of the connections.

The detection rate for each participant is given in Table 10.4. An interesting observation is that there are persons who have a very nice way of drawing, which is easy to be recognized. Other users tend to produce gaps in-between the strokes or draw very rough quadrangles, resulting in wrong recognition of the quadrangles and all corresponding connections, as illustrated in Fig. 10.6d.

As the interactive system acknowledges correct detected shapes, the user of the system has the chance to correct misinterpreted drawings before triggering a search. It is a good

Table 10.4: Detection rate for each participant

Proband	**Quadrangles**	**Adjacent C.**	**Direct C.**
1	1.0	1.0	0.97
2	0.96	0.82	0.81
3	0.93	0.82	0.77
4	0.91	0.95	0.74
5	0.99	1.0	0.87
6	1.0	0.9	0.97
7	0.97	0.95	0.83
8	0.97	0.95	0.84
9	0.99	0.95	0.94
10	0.97	0.91	0.84

result that only one out of twenty rectangles needs to be corrected, making the sketch recognition already practically useful.

We have also performed experiments querying for reference floor plans using our graph search algorithm described in Section 10.2.3. The main complexity is the generation of the decision tree. Our optimization of the algorithm lead into a decrease of the tree nodes from 3.10×10^{31} to 1.09×10^{13}. More details are reported in Weber et al. [209].

10.4 Conclusion and Future Work

In this chapter we have presented an intuitive system for searching floor plans by using a sketch-based interface. Furthermore, a complete system for automatic floor plan analysis is presented which is able to extract structural and semantic content of floor plan from given image. To represent the content of floor plan a graph-based semantic structure and associated visual query language is also presented in this chapter. The recognition rates of sketch recognition as well as floor plan analysis are already good for the use in practice.

Our floor plan analysis system has been evaluated on a database from the literature. We outperform previous state-of-the-art methods and achieve a perfect recognition rate on several documents. Our experiments have shown that the proposed method works very well on a large corpus of 90 floor plans. However, in practice more different types of floor plans exist. We will adopt our methods to other types of plans in our future work.

To improve the detection rates of sketches, a dynamic programming approach to combine

strokes [173] or using a combination of an online and offline detection can be used. Further work for the interactive system will be to offer the architect the freedom to choose between the proposed visual query languages or let him sketch an initial floor plan with his symbolism and extract the semantic graph structure directly from this sketch.

Possible improvements for floor plan analysis consist of normalization of the contours and removing graphical elements outside the outer walls. Furthermore, a weak point of our approach is that it is only able to find the physical existent rooms. This means that if there is a large region and there is no wall with in this region, it will be marked as single room. However, architects tend to divide those rooms still into several functional rooms. This division can be achieved by a more sophisticated semantic analysis based on room labeling results.

In summary, the a.SCatch system is a successful approach integrating state-of-the-art offline image processing and online sketch recognition technologies. Together with the use of recent progress in knowledge management, it provides a novel powerful tool for sketch-based document work, which can be applied to other application areas as well.

11

A Novel Framework for Online Signature Verification

This chapter presents a framework for real-time online signature verification scenarios. The main motivation of this work is to take signature verification to the most commonly occurring real world scenarios, particularly in industry, where signature verification is required. One of the important markets where signature verification is highly demanded is financial institutions. This section describes the application the proposed signature verification framework in different real world scenarios in connection with the Anoto digital pen.

Figure 11.1 illustrates the hardware layout of the Anoto digital pen. This pen specializes in providing the look and feel of regular pens. It only demands to add Anoto dot pattern to any paper and data can be digitized seamlessly. The Anoto pattern makes it possible for the Anoto pen's built-in camera to detect strokes and record signatures that then can be stored in an internal memory or sent via communication unit using Bluetooth/USB. Due to this ease of use, Anoto pens are finding applications in fields from health care to finance. The proposed signature verification framework is an attempt to take signature verification to every area where the Anoto pen finds an application. In particular, it has already been applied in test scenarios for financial institutions and product manufacturing companies.

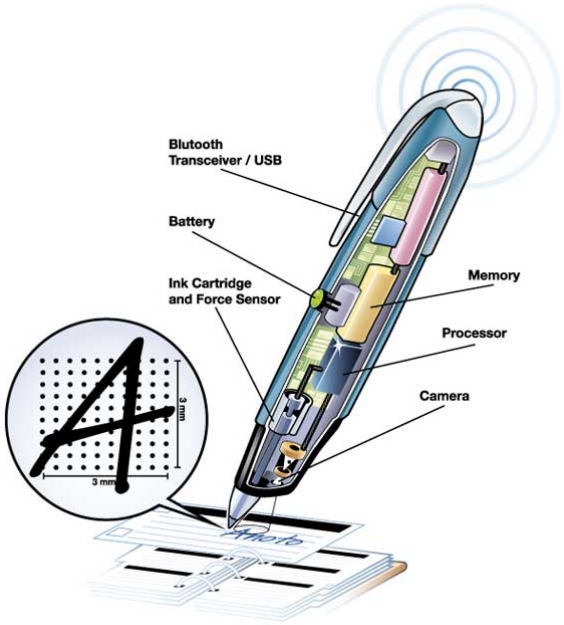


Figure 11.1: Anoto digital pen.

11.1 Signature Verification Framework Overview

The general overview of the signature verification framework is illustrated in Figure 11.2. The online data are collected using the Anoto digital pen and saved in the pen's memory. The pen is then synchronized with some processing device like a computer or a mobile phone. Through this synchronization, data are sent to the Anoto Software Development Kit (SDK). Once the data are received at the SDK, the proposed framework picks the corresponding signatures data (questioned or referenced) and passes it to the signature verification module. The signature verification module then uses a Gaussian Mixture Model (GMM)-based approach to process the signature.

There can be two situations in the framework. In the first situation the user is interacting with the framework for the first time. In this case (s)he has to provide her/his genuine signatures as the reference signatures and prove the identity by any other traditional secure way. Now, the framework generates reference GMMs for the user and stores them. In this way registration of a user is completed.

In the second situation a user interacts with the framework by providing her/his sig-

[0]This chapter is an adapted version of the work presented in Malik et al. [212] "A signature verification framework for digital pen applications" In *DAS* 2012

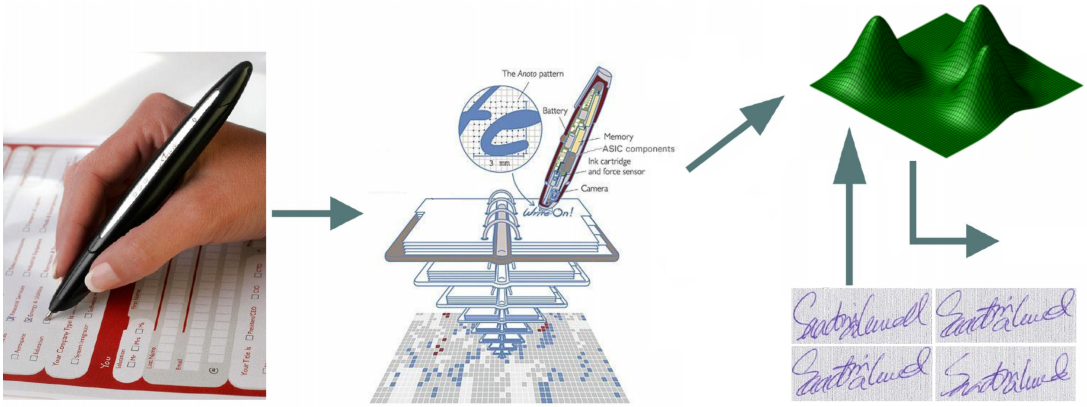


Figure 11.2: General overview of the proposed signature verification framework.

natures and claiming to be some specific person. Now the framework takes the claimed person's GMMs (assuming that this person is already registered with the framework) and fits the questioned person/signature model. Currently, the system reports it's evaluation result in the form of probability values. Based on this value it can be decided whether the claiming person is the authentic writer or a forger.

11.2 Signature Verification Module

The signature verification system used for the framework is an adapted version of a previous system introduced by Liwicki et al. [213]. The basic details are given here only for completion, please refer to [213] for a complete description of the system. Given the online data as an input, the signatures are corrected with respect to their skew and then the following features are extracted; the pen-up/pen-down feature *(1)*; the pressure *(2)*; the speed *(3)*; the speed in x and y direction *(4,5)*; the acceleration *(6)*; the acceleration in x and y direction *(7,8)*; the log radius of curvature *(9)*; the normalized x- and y-coordinate *(10,11)*; the writing direction *(12,13)*; the curvature *(14,15)*; the vicinity aspect *(16)*; the vicinity slope *(17,18)*; the vicinity curliness *(19)*; the vicinity linearity *(20)*; the ascenders and descenders in the off-line vicinity of the considered point *(21,22)*; and the context map, where the two-dimensional vicinity of the point is transformed to a 3×3 map and the resulting nine values are taken as features *(23-31)*.

Gaussian Mixture Models (GMM) have been used to model the signatures of each person. More specifically, the distribution of feature vectors extracted from a person's handwriting is modeled by a Gaussian mixture density. For a D-dimensional feature vector denoted

as x, the mixture density for a given writer (with the corresponding model A) is defined as:

$$p(x\|A) = \sum_{i=1}^{m} w_i p_i(x)$$

In other words, the density is a weighted linear combination of M uni-modal Gaussian densities, $p_i(x)$, each parametrized by a $D \times 1$ mean vector, and $D * D$ covariance matrix. For further details, please refer to [214].

11.3 Application Scenarios

11.3.1 Automatic Order Processing

Highly customized products having shorter development life cycles is the demand of today's global market [215]. This makes efficient order processing an important area for any manufacturing company to improve. In traditional order processing, a client fills an order form and then posts/faxes it to the company. On receiving the order, the company follows its predefined procedure to establish authenticity of the order. One important modality in this process is using the signatures of the client and keeping them in the company's record. This is a time consuming process. Alternatively, web based forms can be used. However, web based order processing suffer from a compendium of difficulties for customer as explained by Doyle et al. [216].

To cope with this an approach for intelligent digital pen-based ordering has been recently introduced Koessling et al. [217]. Here, instead of traditional paper or Internet, a digital pen is used to take customer specific orders that are afterward used in production automation. The customer fills this form as a regular form and signs it. The pen is synchronized (attached to a computer via USB or Bluetooth) and the corresponding electronic form is mapped to the writing information. The final order is then sent to the $SmartFactory^{KL}$ which allows for product automation (further details are provided in [217]).

A pen based interaction order form is shown in Figure 11.3. Here a user may select a product with different colors and enter her/his particulars using the Anoto pen. Note the signature field is also provided as it is on the traditional forms but now it is dealt with the proposed signature verification framework. The paper used here is exactly the same as it was in the traditional approach except that now it also carries the Anoto dot

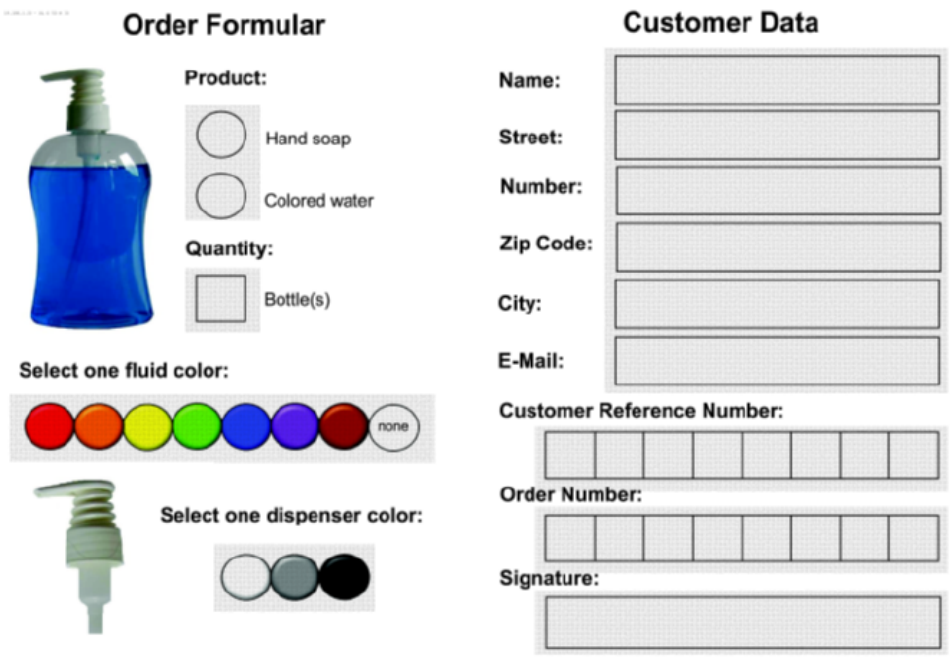


Figure 11.3: A pen based interaction order form.

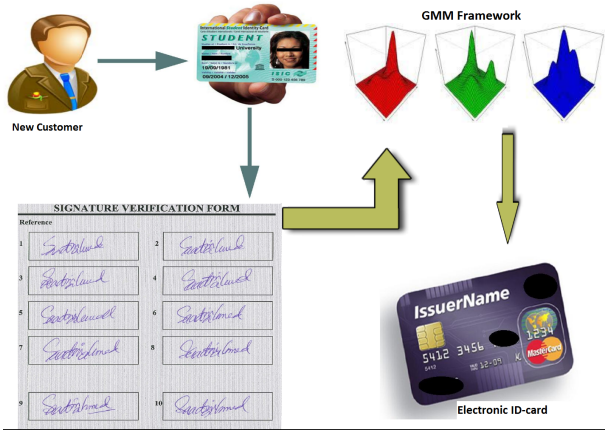
pattern.

The final accept or reject of an order would now depend on the result of the signature verification module. If sufficient likelihood for authenticity can be established, the customized order is sent to the *SmartFactory*^{KL} and the customized production process is started. Otherwise, the order is rejected.

11.3.2 Signature Verification in Banks

Financial institutions bear substantial losses on account of insufficient signature verification mechanisms, where often, a visual comparison is performed to authenticate signatures. This section suggests the idea of integrating the GMM descriptions produced by the proposed framework into electronic ID-cards so that to help banks increase their immunity against fake credit or debit card users.

Two example application scenarios for the framework are illustrated in Figures 11.4(a) and 11.4(b), respectively. Figure 11.4(a) illustrates a scenario where a customer registers at a bank for the first time. At first the customer applies for opening an account by providing her/his identity. This may be done by any card containing picture and



(a)

(b)

Figure 11.4: Application scenarios of the proposed framework: (a) registering a customer and generating an electronic ID-card; (b) establishing the authenticity of a customer.

particulars of the applicant like a passport or personal ID-card. For opening an account the bank takes a certain number of online signatures from the customer and provides them to the framework proposed here. The framework is then applied to take GMM descriptions using various combinations of these genuine reference signatures. Note that already during this process a validation can be performed that all signatures are similar enough to produce a probability of being authentic. The obtained GMM descriptions are then stored on the electronic card of the customer. By doing this, the original signatures would not be available on the card (protecting the customer against fraud). Only the model used for comparison will be available. Furthermore, one may not generate the genuine signatures from the stored model thereby removing the danger of any security

attack of such kind.

Now the customer, whenever using the card for monetary transaction may use his/her signatures instead of or additionally to pin codes/logins. Figure 11.4(b) illustrates this scenario. Here a customer has an electronic card (developed in the previous scenario) after purchasing goods at a supermarket tries to pay with this card. At the counter (s)he provides her/his electronic card along with signatures written with a digital pen. These signatures are transferred to a local computer having an instance of the framework where the authenticity of these signatures is judged. If the framework then reports an accept it refers that the customer is authentic. Otherwise, the customer might be a forger/imposter trying to use some other person's card (in which case, another authentication method can be applied). There can also be various other applications of these *behavioral information containing electronic cards* which are beyond the scope of this discussion.

11.4 Datasets and Evaluation

The evaluation of the framework was performed on two data sets. The first data set contained the data collected specifically using the Anoto digital pens on forms having Anoto dot pattern. For this collection, ten authors (male and female) from different countries aging between 18 to 40 provided their genuine signatures. Ten forgers (students, researchers, and a calligrapher) were asked to make skilled forgeries of each genuine author. Each genuine author contributed 9 of her/his signatures. Out of these nine, 7 signatures were used as reference signatures and remaining 2 were put in the test set for

Table 11.1: Evaluation results of Anoto online data (data set 1)

Author-ID	FA	FR
1	0	0
2	0	0
3	2	1
4	0	1
5	0	1
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

every genuine author. Each forger also produced 9 forgeries. As a whole, the test set for this data set contained 20 genuine signatures and 90 skilled forgeries.

The second data set was the NISDCC signature collection of the ICDAR 2009 online signature verification competition (SigComp)[1]. This data set consists of 60 authentic signatures written by 12 authors, 31 forgers produced skilled forgeries at the rate of 5 forgeries per genuine author.

Since the number of signatures in the first data set is quite low, the results are reported in terms of number of Falsely Accepted signatures (FA) and number of Falsely Rejected signatures (FR). The evaluation results indicate initial success that is also triggering the interest of industry (financial institutions) in this area.

[1]publicly available for research purposes at http://sigcomp09.arsforensica.org

12

A Part-based Approach for Signature Verification

Automatic signature verification is needed in various different areas of our daily life. These include banks, governmental, security, and financial institutions, etc. Since the last few decades, researchers from Pattern Recognition (PR) community are developing and continuously improving different automatic signature verification systems for both offline (where only spatial information of signatures is available) and online (where both spatial and temporal information of signatures is available) cases. In either case, (online or offline), the verification problem is usually solved by classifying signatures into two classes, i.e., either as genuine or forged. This classification is helpful in many cases, but in some areas, e.g., forensic handwriting/signature analysis, another important genre of signatures, i.e., disguised signatures, needs classification. Although in forensic examination, disguised signatures are of high importance [218], yet they are often neglected by PR researchers [219].

Disguised signatures are usually difficult to identify [219], as they are written by the genuine author but with intention to deny its authorship. This could be done mainly for fraud e.g., a disguised signature made on a bank check can be used to withdraw cash and later on it can be claimed that the check did not contain original signatures. In such a case, it is very difficult for bank/financial institution officers to distinguish between genuine and disguised signatures. In addition, it is also not possible to have an expert forensic examiner available in all of the institutes, which require authentication via signatures, who can first analyze signatures and then allow the next step in the routine work flow. It is, therefore, required to enable automatic systems classify the three different genres of

[0]This chapter is an adapted version of the work presented in Malik et al. [97] "FREAK for Real Time Forensic Signature Verification" In *ICDAR* 2013

signatures, i.e., genuine, forged, and disguised, so that different types of such frauds can be prevented.

This chapter presents a novel method for automatic signature verification, which is able to deal with genuine, forged, and disguised signatures at the same time, with comparatively low EER and very high time efficiency. The presented method is based on part-based analysis of signature images. In particular FREAK features are used which are inspired by human visual perception. In addition to EER, the presented method is also computationally efficient and only requires 0.6 seconds, on average, to verify an offline signature of dimensions nearly 3000×1500. This shows that there is a strong potential for using the presented method in various real time scenarios, such as the bank scenarios noted earlier. Furthermore, the presented method is evaluated on the publicly available dataset of 4NSigComp2010 (the first ever signature verification competition with data containing disguised along with genuine and forged signatures). This data is collected by Forensic Handwriting Examiners (FHEs) from forensic-like situations [219].

The rest of the chapter is organized as follows. Section 12.1 summarizes different automatic verification systems available for signatures. Section 12.2 provides details about the presented system for forensic signature verification. Section 12.3 details the dataset used for evaluation. Section 12.4 presents evaluation results in terms of EER, and execution speed on the said publicly available dataset. Finally, Section 12.5 concludes the chapter and provides hints for possible future improvements.

12.1 Related Work

Signature verification has remained an active field in the last few decades. The recent state-of-the-art of signature verification is summarized in [220]. Nearly all of the state-of-the-art methods have been tested for detection of genuine and forged signatures but disguised signatures are generally neglected, apart from some initial research, like [221], and in some comparative studies of local and global feature based methods, like [222].

Recently, a system for online forgery and disguise detection has been developed which combines online signature features through several classifiers [223]. This system, however, is online and forensic experts are usually interested in offline automatic verification. Furthermore, it does not suit the disguise signature classification scenario on bank checks since the bank check disguise can often occur when the original author is not physically

present there, otherwise if s(he) is present there then s(he) may not disprove this fact later as there may be other witnesses. Unlike disguised signatures, disguised handwriting in general is previously considered in some PR-research like [224]. However, [224] only focuses the disguise and genuine handwriting and this does not completely suffice the needs of handwriting experts.

The FAST keypoint detector, which was used to initially identify the signatures' local regions of interest, has been previously used mainly for problems like multiple object tracking [225], object recognition for smart phone platforms [226], and recognition of degraded handwritten characters [227]. The SURF Keypoint detector, which were also used to initially identify the signatures' local regions of interest in one of our experiments, in conjunction with SURF keypoint descriptor has been previously used heavily for object and character recognition, such as in [8, 55, 228].

The novelty of our work is the explicit use of FAST and SURF keypoint detectors and FREAK local features for offline verification to cater the complete signature verification paradigm including disguised signatures. To the best of the authors' knowledge, the presented system is the first automatic system combining the traditionally known keypoint detectors and descriptors (FAST + FREAK, and SURF + FREAK) for the purpose of signature verification and also reporting the efficiency of these systems in terms of time.

12.2 Methodology

The methodology for the presented signature verification is based on part-based/local features. To perform part-based analysis, it is first required to extract keypoints from the signature images. The regions around these keypoints are then described using different descriptors. Note that different feature detectors and descriptors can be used individually or in a combined fashion to detect local keypoints and describe them. Hence, in the presented methodology, FAST [1] keypoint detector is used to detect keypoints in signature images. FAST keypoint detector is computationally efficient in comparison to well known keypoint detection methods, e.g., SIFT [58], Harris [64], and SURF [103] (results are also provided when SURF keypoint detector is used for finding the potential local areas of interest). In addition, FAST gives a strong response on edges, which makes it suitable for the task of signature verification. For further details about FAST, please refer to Section 3.2. Once the keypoints are detected, descriptor for each of the keypoints is com-

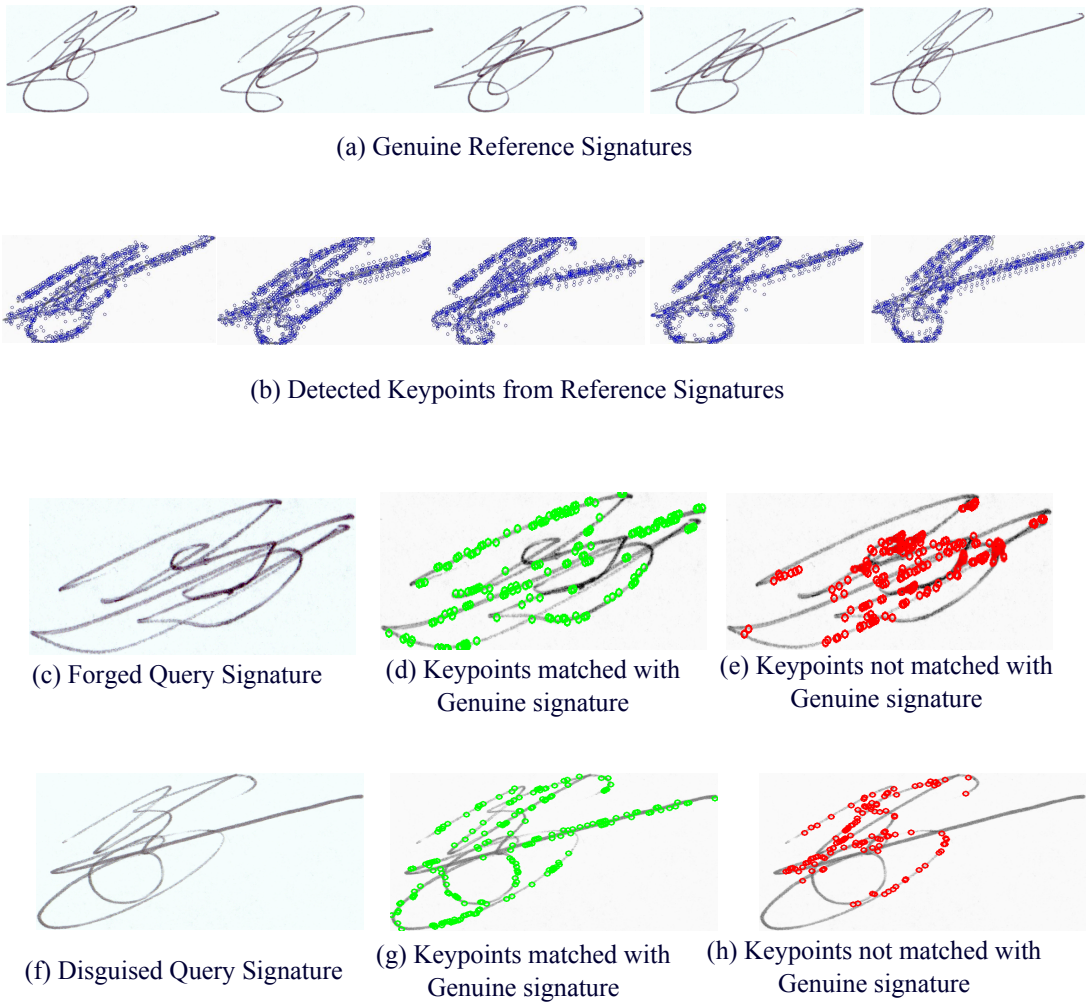


Figure 12.1: Steps of signature verification. Green dots: keypoints of the query/questioned signature that matched with the keypoints of the reference signatures. Red dots: keypoints of the query/questioned signature that did not match with the keypoints of the reference signatures.

puted using a part-based descriptor, FREAK [3]. FREAK is a binary keypoint descriptor which is computationally very efficient. For further details about FREAK, please refer to Section 3.6. As the descriptors extracted using FREAK are binary, therefore Hamming distance is used for comparison of descriptors of query and reference signatures. The use of Hamming distance in-turn makes it computationally more efficient as it can be computed using a simple XOR operation on bit level.

To categories a signature as genuine, forged, or disguised, first it is binarized using the well known global binarization method OTSU [229]. The OTSU binarization is chosen since the data had fairly high resolution signature images and OTSU is also computationally

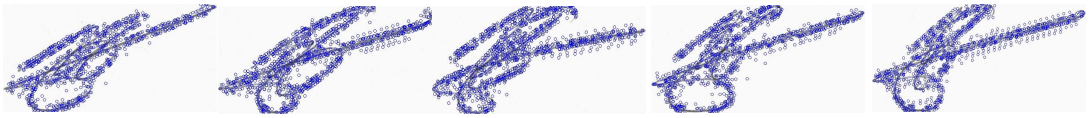
efficient. After binarization, the following procedure is followed.

1. Apply the FAST or SURF keypoint detector on all the reference signatures, separately, to get the local areas of interest (keypoints) from these signatures.
2. Then, get the descriptors of all of these keypoints present in all reference images using the FREAK keypoint descriptor, which describes each keypoint with a 64 bit descriptor.
3. All of these keypoints and their associated descriptors describing local information are added into a database. The database thus contains features for all of the keypoints which are collected from all reference signature images.
4. Once the features database is created, keypoints and descriptors are extracted for the query/questioned signature.
5. Now a comparison is made between the query signature keypoints and the keypoints present in the features database for each corresponding author.
6. The same process of detecting local area of interest using FAST and then descriptors by FREAK is applied to the query image.
7. Take the first keypoint of the query Image and compare it with all the features present in the features database, one by one. If a query signature keypoint is at a distance less than θ, from any feature present in the features database, note the keypoint.
8. Keep this process going until all the query signature's keypoints are traversed.
9. Calculate the average by considering the total number of query keypoints and the query keypoints matched with the features database. This represents the average local features of the questioned signature that are present in the database of that author.
10. Now, if this average is greater than the threshold θ, (meaning, most of the questioned signature local features are matched with reference local features), the questioned signature is be classified as belonging to the authentic author, otherwise if this average is less than the threshold θ, (meaning, there are only a few query keypoints for whom any match is found), it does not belong to the authentic author.

The same procedure is followed when using the SURF or the FAST keypoint detector. Classification in both the cases is based on the local descriptions provided by FREAK

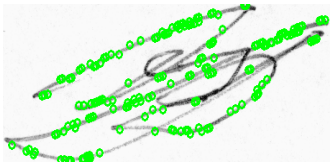
features.

(a) Genuine Reference Signatures

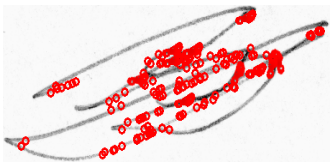


(b) Detected Keypoints from Reference Signatures

(c) Forged Query Signature

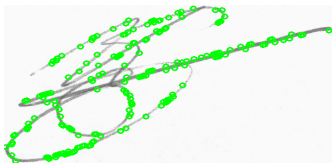


(d) Keypoints matched with Genuine signature

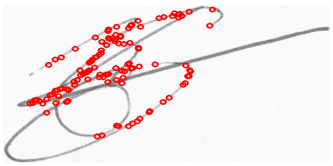


(e) Keypoints not matched with Genuine signature

(f) Disguised Query Signature



(g) Keypoints matched with Genuine signature



(h) Keypoints not matched with Genuine signature

Figure 12.2: Steps of signature verification. Green dots: keypoints of the query/questioned signature that matched with the keypoints of the reference signatures. Red dots: keypoints of the query/questioned signature that did not match with the keypoints of the reference signatures.

Figure 12.2 provides some example signatures where the above-mentioned methodology is applied. Figure 12.2 (a) shows some sample genuine signatures and the related Figure 12.2 (b) shows the keypoints extracted from the reference signatures using FAST feature detector. Figure 12.2 (c) shows a questioned signature that in fact is a forgery attempt. Figure 12.2 (d) and (e) provide green and red keypoints on this forgery attempt. The green keypoints are those which belong to the query signature but they also matched with the keypoints present in the features database. The red keypoints are those which belong to the query signature but they did not match with the features database. Similarly,

Table 12.1: Summary of the comparisons performed between the presented systems, SURF-FREAK, FAST-FREAK and the participants of 4NSigComp2010.

System (sec.)	FAR	FRR	EER	Time (sec.)
1	1.1	90	80	312
2	41.1	90	58	1944
3	20.0	70	70	85
4	0.0	80	70	19
5	13.3	80	55	45
6	87.0	10	60	730
7	1.1	80	70	65
(SURF-FREAK)	30	30	30	12
(**FAST-FREAK**)	**30**	**30**	**30**	**0.6**

Figure 12.2 (f), (g), and (h) are obtained when the query signature was a disguise attempt. From the Figure 12.2 (c through e), one can see that for lines which were more or less straight in nature there was a high match of query signature with the reference signature, even if the query signature was a forgery attempt. However, for curved lines/strokes (which are usually very specific of particular individuals) a large number of keypoints mismatched. For disguised query signature, Figure 12.2 (f through h), larger number of keypoints matched with the reference signatures. This is because the disguise signature belonged to the original/authentic author who left some traces of her/his identity while trying to disguise. This can be attributed to the brain motor activity which is always utilized whether a person is trying to disguise or not.

12.3 Dataset

We used the evaluation set of the 4NSigComp2010 signature verification competition. This is the first ever publicly available dataset containing disguised signatures. The collection contains only offline signature samples. The signatures are collected by forensic handwriting examiners and scanned at 600 dpi resolution. The collection contains 125 signatures. There are 25 reference signatures by the same writer and 100 questioned signatures by various writers. The 100 questioned signatures comprise 3 genuine signatures written by the reference writer in her/his normal signature style and 7 disguised signatures written by the reference writer where s(he) tried to disguise herself/himself (the reference writer provided a set of signatures over a five day period); and 90 simulated signatures (written by 34 forgers freehand copying the signature characteristics of the reference writer. The forgers were volunteers and were either lay persons or calligraphers.).

All writings were made using the same make of ball-point pen and using the same make of paper.

We used the same experimental protocol as was used in the 4NSigComp2010. For training, 25 genuine reference signatures were available and after training on these 25 genuine reference signatures our systems had to classify correctly the 100 questioned signatures (the test set). Note that, no forgery was used for training and yet our systems were able to classify. This is a realistic forensic scenario [219] and is posed in the 4NSigComp2010.

12.4 Evaluation

Two experiments were performed for evaluating the efficiency and performance of the presented systems.

- Experiment 1 that focused the classification of disguised, forged, and genuine signatures using FAST Kepypoint detector and FREAK features.
- Experiment 2 that focused the same classification using SURF Kepypoint detector and FREAK features.

The results of these experiments are provided in Table 12.1. As shown in the table, both of the newly presented systems, i.e., SURF-FREAK and FAST-FREAK outperform all the participants of the 4NSigComp2010 signature verification competition. The best system from the competition could achieve an EER of 55% in the presence of disguised signatures in the test set. Both of the presented systems, however, achieve an EER of 30% which is remarkable when compared to the other systems.

Furthermore, Table 12.1 also presents the performance comparison of the said systems on the basis of time. The time is given in seconds and is actually the average time taken by any algorithm to report its result on the authenticity of one questioned signature. For reporting this result, the system has to process the questioned as well as 25 reference signatures. Both of the presented systems again outperformed all the participants. Specially the FAST-FREAK method is extremely time efficient. It succeeds from the other eight methods by a times of (520, 3240, 141, 31, 75, 1216, 108, and 20, respectively). All tests were performed at a machine with the following specifications.

- Processor: Intel Dual Core 1.73 GHz

- Memory: 1GB
- OS: WinXP Professional

The presented methodology is generic in nature as different combinations of local feature detectors and descriptors can be used. Two systems are presented which follow the same underlying methodology, one based on SURF detector with FREAK descriptors, and the other based on FAST detector with FREAK descriptors. Other combinations of detectors and descriptors could also be applied likewise. The results indicated that the presented methodology fairly suits local feature based approaches and beats the state-of-the-art by a large margin, both in term of time and error rate. Furthermore, a general drawback of most of the local features based approaches is the enormous amount of time they take to compute results. By using the presented methodology, the presented systems were extremely efficient compared to other systems who were mostly relying on global features [219]. This shows that, if utilized properly, local feature approaches show the potential of improving both performance and efficiency of classification.

12.5 Conclusion and Future Work

This chapter presents a novel part based system for forensic signature verification involving disguised signatures. Experiments are performed with two different local feature detectors, namely SURF and FAST. In both of these cases FREAK features are used and applied our novel methodology to classify genuine, forged, and disguised signatures. Both the presented systems outperformed all the participants of the 4NSigComp2010 signature verification competition by achieving an EER of 30%. Whereas the EER of the best participant in the said competition was 55%.

Furthermore, a time efficiency comparison is made between the presented methodology and the participants of the 4NSigComp2010. Although, the presented approach is based on local feature analysis, yet it is capable of performing classification with better EER and manifold faster execution.

In future it is planned to use larger data sets where disguised signatures from large number of authors are present in the test set. Regarding the systems' outcomes, it is planned to enable them produce likelihood ratios according to Bayesian approach, which will make these systems even more useful in the real world forensic casework. This, however, is a

difficult task since respective likelihood computation of multiple classes is required in this case.

Part V

CONCLUSION

13 Conclusion and Future Work

This chapter summarizes the major conclusions that could be drawn from the issues considered in this thesis and the solutions presented to tackle them. Any limitations of these solutions are discussed along with the possible research that would overcome these limitations in the future.

13.1 Conclusions

The aim of this thesis is to address the problem of information segmentation in document images. A document image contains different types of information, for instance, text (machine printed/handwritten), graphics, signatures, and stamps. It is necessary to segment information in documents so that to process such segmented information only when required in automatic document processing workflows. The main motivation for this thesis comes from the author's observation that despite of their immense need in research and industry, there is a very limited, availability of robust automatic information segmentation systems. While addressing the problem of information segmentation, it was observed that different types of information available in documents have some very specific and unique properties (e.g., texture, intensity gradient), which could be utilized to differentiate them from one another. Humans are very good at differentiating objects based on these properties. The aim is to develop automatic methods, which could identify and learn these properties and use them to segment different types of information available in document images.

The main contribution of this thesis is the conceptualization and implementation of an

information segmentation framework that is based on part-based features. The generic nature of the presented framework makes it applicable to a variety of documents (technical drawings, magazines, administrative, scientific, and academic documents) digitized using different methods (scanners, RGB cameras, and hyper-spectral imaging (HSI) devices). A highlight of the presented framework is that it does not require large training sets, rather a few training samples (for instance, four pages) lead to high performance, i.e., better than the previously existing methods. In addition, the presented framework is simple and can be adapted quickly to new problem domains. In this thesis, the presented framework is applied on several problems and evaluation results show that the approaches based on the presented framework work better than the previously existing methods for respective problems.

In the area of scanned document images, the presented framework is used to solve three different types of information segmentation problems. First, based on the information segmentation framework, this thesis presents a novel system for signature segmentation in administrative documents. These documents often contain information other than just the signatures, e.g., background text, lines, and logos. In order to authenticate these documents by signature verification, the signatures must firstly be segmented and extracted. The presented system uses SURF as part-based features for both keypoint detection and feature extraction. The segmentation system is tested on the Tobbaco-800 dataset (a publicly available dataset) where it outperforms the state-of-the-art methods by extracting all of the available signatures.

The second contribution in the area of scanned document images is segmentation of stamps in administrative documents. Stamps also serve as a seal of authenticity for documents. However, the location of stamp on a document can be more arbitrary than signatures depending on the content of a document and the person sealing that document. A novel system based on the generic information segmentation framework is presented for stamp segmentation where FAST based keypoint detection is combined with BRIEF (binary descriptor) features. It is able to segment stamps of all categories, i.e., textual, graphical, regular shaped, and irregular shaped. In addition, the presented method is able to detect colored as well as black stamps. The main highlight of the presented method is segmentation/extraction of black stamps, as the existing methods for stamp detection are not able to segment black stamps since they are based on the assumption that stamps are colored objects. The evaluation results show that the presented system achieved recall and precision of 75% and 84%, respectively and outperforms the state-of-the-art

especially for black stamps.

The third contribution in the area of scanned document images in the domain of information segmentation in technical drawings, e.g., architectural floorplans, maps, and circuit diagrams. A system based on the presented information segmentation framework is developed for extracting text components touching graphics. The presented system while tested on a publicly available dataset of architectural floorplans extracted more than 95% of the characters which were touching graphics components. In fact, these extracted characters were actually the problematic characters for the previously existing text/graphics segmentation methods. This method increases the overall recall of the existing methods, as the touching text components which are not extractable by existing methods can be easily extracted by the presented method.

The contributions of signature, stamp, and text touching graphics segmentation validate our hypothesis that "it is possible to develop a generic part-based approach, which is capable of segmenting different types of information in document images". This is because three different types of information are segmented using systems based on the presented framework. In addition, it validates another hypothesis that "the part-based approach can be adapted to fulfill task-specific computational and performance requirements" as a combination of FAST keypoint detector with BRIEF features results into a system which is able to segment stamp from 300 dpi document image in almost 16 seconds where SURF based system needs more than 150 seconds to segment a stamp.

In the area of HSI document images, two contributions have been made. First, this thesis presents a system for signature segmentation in hyper-spectral document images, commonly used by FDEs. In HSI the color spectrum (starting from ultraviolet, including visible, and ranging to the infrared region) is divided into a large number of bands which result in a very fine and detailed representation compared to the 3-channel RGB model. A novel system for automatic signature segmentation from HSI document images is introduced. The presented system is also based on an adapted version of the generic information segmentation framework presented in this thesis. This system is quite unique as it is the first method, ever reported for segmentation of signatures from documents, combining part-based features with hyperspectral imaging. It is different from all existing signature segmentation methods as it does not use any structural information but only the hyperspectral response of the document, regardless of different colors used in the document. The proposed method is also independent of the type, and density of the ink used for writing signatures. A very important aspect of the proposed method is that its

performance is consistently very good even when the signatures are overlapping with text or other information available in the document, e.g., tables, printed text, stamps, logos, and so on. Furthermore, a new dataset comprising of 300 documents captured using a high-resolution hyper-spectral scanner is presented. Evaluation of the presented method on this hyper-spectral dataset shows that it is able to extract signature pixels with the precision and recall of 100% and 79%, respectively. The contributions in HSI documents validate our hypothesis that "the part-based approach can be leveraged to cope with hyper-spectral document images".

In addition to information segmentation, this thesis also addresses the problem of automatic ground truth generation in camera-captured document images. This thesis presents novel, generic method (based on part-based features) for automatic ground truth generation of camera-captured document images (books, magazines, articles, invoices, etc.). It is fully automatic and does not require any human intervention for labeling. Evaluation of the sample from generated datasets shows that this system can be successfully applied to generate very large-scale datasets automatically (which is not possible via manual labeling). While comparing the proposed ground-truth generation method with humans, it was revealed that the proposed method is able to label even those words where humans face difficulty in reading due to bad lighting and/or blur in the image. The proposed method is generic as it can be used for generation of dataset in different languages (English, Russian, etc.). In addition, it is not limited to camera-captured documents and can be applied to scanned images. This contribution validates our hypothesis that "the use of part-based approach facilitates the automatic generation of large-scale ground-truth for related document image analysis tasks". Furthermore, a large dataset (called C^{3}Wi) of camera-captured characters and words images, comprising 1 million word images (10 million character images), captured in a real camera-based acquisition.

A novel deep learning based system for the recognition of camera-captured document images is also presented in this thesis. The proposed character recognition system is able to learn from large datasets and therefore trained on C^3Wi dataset. Furthermore, various benchmark tests are performed to uncover the behavior of commercial (ABBYY), open source (Tesseract), and the presented camera-based OCR using the presented C^{3}Wi dataset. Evaluation results reveal that the existing OCRs, which already get very high accuracies on scanned documents, have limited performance on camera-captured document images; where ABBYY (75%), Tesseract (50.22%), while the presented character recognition system has an accuracy of 95.10%. This contribution validates our hypothesis

that "a large dataset enables to perform deep learning even when the ground-truth has been generated automatically and contains errors".

In addition to the above-mentioned contributions, this thesis also looked into areas closely related to information segmentation and application of part-based features. In this context, this thesis presented a system for automatic analysis and sketch based retrieval of architectural floor plans. This system is a perfect example of systems, which need segmented information at different points in time during analysis. In addition, this thesis also presents a novel part-based approach for offline signature verification system and a novel generic framework for online-signature verification system. These contributions also confirm that part-based features can be leveraged to build systems, which are capable of performing different tasks, e.g., segmentation, recognition, and verification.

13.2 Limitations

This section describes the limitations of different methods presented in this thesis.

First this thesis presented a generic framework for information segmentation in variety of documents (technical drawings, magazines, administrative, scientific, and academic documents) digitized using different methods (scanners, RGB cameras, and hyper-spectral imaging (HSI) devices). The presented information segmentation framework is generic in a sense that it can be used to segment different types of information from different types of documents. The framework can be used directly for most of the formal documents, e.g., administrative documents, invoices, and utility bills. This is because the type of information, which will potentially appear in these documents, is known in advance. For example, it is already known that a utility bill will contain printed text, logo, and stamp. The presented framework can be used in all such scenarios where type of information, which will potentially appear in these documents, is known in advance. However, a limitation of the presented framework is that it is based on an assumption that different types of information, which are going to appear in a document, are known in prior. This is mostly true for the formal documents, e.g., invoices, utility bills, deposit slips, legal notices, and formal letters. However, this is not valid in case of informal documents e.g., personal notes, recipes, where it is unknown that what type of information can appear in a document.

The presented framework includes some thresholds which varies and need to be adjusted

based on the target application. More specifically, it is required to define a value of similarity threshold for performing feature selection, where similar features are removed from the databank. This threshold varies with the choice of feature descriptors and application specific requirements for precision and recall. The threshold dependency is also a limitation of the presented information segmentation framework.

In this thesis signature segmentation is also performed in special type of documents, i.e., HSI documents. The use of HSI for signature segmentation though seems very promising, yet it has an inherent limitation, i.e., currently the HSI devices are too expensive and are not available easily in many areas of the world. Nonetheless, the author hopes that with time this technology will get cheaper and will benefit many areas of document analysis especially signature segmentation and verification.

In addition to information segmentation framework, this thesis also provides a part-based generic method for automatic ground-truth generation for camera-captured document images. The presented method is generic and can be used to build large-scale labeled dataset of camera-captured document images in different languages. The presented method performs very well on text documents, e.g., books, magazines, and newspapers and is able to generate ground truth for them. However, this method also has some inherent limitations. Although it is a limitation, a main characteristic is that it requires a PDF of the captured document, so that it can extract and label the captured image using the text from PDF. This PDF is not available in many cases, e.g., for historical documents. In addition, the method is based on LLAH, which works well on the text dominant documents and have limited performance on natural scene images. Therefore, despite being generic in the sense that it can be used to generate dataset of different languages but this method to text dominant documents having PDF files available beforehand.

13.3 Future Work

This section provides an overview of various research dimensions to consider for the future.

First, to further solidify the findings of this thesis, it is planned to build a large dataset specifically designed for the task of information segmentation and verification. Currently most of the existing signature segmentation systems are evaluated on subsets of Tobacco-800 dataset and (to the best of author's knowledge) there is no publicly available dataset specifically designed for signature segmentation. The major disadvantage with the

Tobacco-800 dataset is that it only contains patch level information about signatures, i.e., which block primarily contains signatures. Another problem in the Tobacco-800 dataset is that images are already binarized and a lot of information important for segmentation is lost. Note that the StaVer dataset is available for stamp segmentation, and verification, which contains both patch and pixel level ground truth information, but only for stamps. There is no ground truth for signatures, logos, and text. In the future, it is planned to build a dataset containing signatures, printed and handwritten text, stamps, and logos, with pixel level ground truth information. The dataset will be collected in a way that it also contains the information about authors who have signed these documents. This will enable the tuning of signature verification systems in such a way so as to enable of distinguish between genuine and forged signatures even in the presence of some noise, e.g., overlapping characters or missing parts of signatures. Furthermore, in the planned dataset, multi-spectral information about signatures and other parts of documents will also be incorporated. It is planned to perform various benchmark tasks using the planned dataset so that systems can be evaluated in terms of recall (how many items are extracted from the document), precision (how many of the extracted items are actually of the target type), and efficiency (both in terms of speed and complexity).

In context of the presented generic information segmentation, it is planned to extend and test the framework for multi-class problems (segmentation of different types of information, e.g., text (machine printed, handwritten), signatures, stamps, logos, and tables present in the same document. To further strengthen the presented information segmentation framework, it is planned to introduce unsupervised learning, especially for the feature-learning step. The use of unsupervised learning will make the presented information segmentation framework independent of any threshold. In addition, it is planned to extend the presented information segmentation to work on other problems as well and not only on traditional document images. This includes segmentation in natural scenes and historical documents.

This thesis also presented novel methods for signatures and stamp segmentation based on the presented generic framework. It is planned to integrate logo detection with the currently presented methods to further increase the precision. In addition, to deal with the overlapping stamps and signature with text, it is planned to integrate the presented method for segmentation of text touching graphics as a post-processing step. Furthermore, the stamp and signature segmentation methods presented in the thesis are orthogonal to the method presented by Micenkova and van Beusekom [19]- so both methods

can be easily combined to get the benefits of both, i.e., first color-based segmentation and then part-based processing.

This thesis also presents a method for automatic ground truth generation of camera-captured document images. Currently, this method is capable of working only on the text dominant document images. A main problem in the recognition of historical document images is the non-availability of ground truth. Now-a-days different archives of historical documents are available on-line with their ground truth information in Text Encoding Initiative (TEI) format. In the future, it is planned to adapt the presented ground truth generation method for automatic ground truth generation of historical documents using TEI information. This will facilitate building large-scale historical documents database which will enable building systems for analysis and recognition of historical documents. In addition, in the future, it is planned to build dataset for different languages, including Japanese, Arabic, Urdu, and other Indic scripts. Furthermore, it is planned to use the presented C^3Wi dataset for domain adaptation. This means that training a model on C^3Wi dataset with the aim to make it working on natural scene images.

cation systems need to be developed to make it a viable cation. Along with patch level information, this dataset will have also cation frameworks for analysis of documents containing signatures. Furthermore, in this

Bibliography

[1] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005.*, volume 2, pages 1508 –1515 Vol. 2, oct. 2005.

[2] Stefan Leutenegger, Margarita Chli, and Roland Yves Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555. IEEE, 2011.

[3] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. Ieee, 2012.

[4] Historical and interesting bank checks.

[5] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):855–868, May 2009.

[6] A. Garg and M. Datar. Document archiving system, July 3 2008. US Patent App. 11/847,055.

[7] John Cullen and Mark Peairs. Document management system, April 13 1999. US Patent 5,893,908.

[8] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417. Springer, 2006.

[9] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[10] Tim Frey, Marius Gelhausen, and Gunter Saake. Categorization of concerns: A categorical program comprehension model. In *Proceedings of the 3rd ACM SIGPLAN Workshop on Evaluation and Usability of Programming Languages and Tools*, PLATEAU '11, pages 73–82, New York, NY, USA, 2011. ACM.

[11] K. Ueda. Extraction of signature and seal imprint from bankchecks by using color information. In *Proceedings of the Third International Conference on Document Analysis and Recognition,*, volume 2, pages 665 –668 vol.2, aug 1995.

[12] A. Soria-Frisch. The fuzzy integral for color seal segmentation on document images. In *Proceedings. 2003 International Conference on Image Processing, 2003. ICIP 2003.*, volume 1, pages I – 157–60 vol.1, sept. 2003.

[13] Liang Cai and Li Mei. A robust registration and detection method for color seal verification. In *Proceedings of the 2005 international conference on Advances in Intelligent Computing - Volume Part I*, ICIC'05, pages 97–106, Berlin, Heidelberg, 2005. Springer-Verlag.

[14] Guangyu Zhu, Stefan Jaeger, and David Doermann. A Robust Stamp Detection Framework on Degraded Documents. In *International Conference on Document Recognition and Retrieval XIII*, pages 1–9. San Jose, CA, 2006.

[15] Li-jiang Chen, Tie-gen Liu, Jia-jia Chen, Jun-chao Zhu, Ji-jie Deng, and She-xiang Ma. Location algorithm for seal imprints on chinese bank-checks based on region growing. *Optoelectronics Letters*, 2:155–157, 2006.

[16] Partha Pratim Roy, Umapada Pal, and Josep Lladós. Seal object detection in document images using ght of local component shapes. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 23–27, New York, NY, USA, 2010. ACM.

[17] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, January 1972.

[18] Dariusz Frejlichowski and PawełForczmański. General shape analysis applied to stamps retrieval from scanned documents. In *Proceedings of the 14th interna-*

tional conference on Artificial intelligence: methodology, systems, and applications, AIMSA'10, pages 251–260, Berlin, Heidelberg, 2010. Springer-Verlag.

[19] B. Micenkova and J. van Beusekom. Stamp detection in color document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1125 –1129, sept. 2011.

[20] R. Jayadevan, S.R. Kolhe, P.M. Patil, and U. Pal. Automatic processing of handwritten bank cheque images: a survey. *IJDAR*, 15:267–296, 2012.

[21] Jayadevan, S. Subbaraman, and P.M. Patil. Variance based extraction and hidden markov model based verification of signatures present on bank cheques. In *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, volume 2, pages 451–455, Dec 2007.

[22] S. Djeziri, F. Nouboud, and R. Plamondon. Extraction of signatures from check background based on a filiformity criterion. *TIP*, 7(10):1425–1438, October 1998.

[23] Vamsi Krishna Madasu, Mohd Hafizuddin, Mohd Yusof, M. Hanm, and Lu Ss. Automatic extraction of signatures from bank cheques and other documents. In *Proceedings of DICTA*, pages 591–600, 2003.

[24] M. Sankari, M. Benazir, and R. Bremananth. Verification of bank cheque images using hamming measures. In *ICARCV 10*, pages 2531 –2536, dec. 2010.

[25] Guangyu Zhu, Yefeng Zheng, David Doermann, and Stefan Jaeger. Multi-scale structural saliency for signature detection. In *In Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8, 2007.

[26] Guangyu Zhu, Yefeng Zheng, David Doermann, and Stefan Jaeger. Signature detection and matching for document image retrieval. *TPAMI*, 31(11):2015–2031, 2009.

[27] Ranju Mandal, Partha Pratim Roy, and Umapada Pal. Signature segmentation from machine printed documents using conditional random field. *ICDAR*, 0:1170–1174, 2011.

[28] Ranju Mandal, Partha Pratim Roy, and Umapada Pal. Signature segmentation from machine printed documents using contextual information. *IJPRAI*, 2012.

[29] Friedrich M. Wahl, Kwan Y. Wong, and Richard G. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20(4):375 – 390, 1982.

[30] Syed Saqib Bukhari, Mayce Ibrahim Ali Al Azawi, Faisal Shafait, and Thomas M. Breuel. Document image segmentation using discriminative learning over connected components. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 183–190, New York, NY, USA, 2010. ACM.

[31] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos. Icdar 2009 page segmentation competition. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 1370–1374, July 2009.

[32] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Improved document image segmentation algorithm using multiresolution morphology. In *Document Recognition and Retrieval XVIII - DRR 2011, 18th Document Recognition and Retrieval Conference, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 24-29, 2011, Proceedings*, pages 1–10, 2011.

[33] Ritu Garg, Ehtesham Hassan, Santanu Chaudhury, and M. Gopal. A crf based scheme for overlapping multi-colored text graphics separation. In *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, ICDAR '11, pages 1215–1219, Washington, DC, USA, 2011. IEEE Computer Society.

[34] Chanchal A Chandrakar Priti P Rege. Text-image separation in document images using boundary/perimeter detection. *ACEEE International Journal of Signal and Image Processing*, 4(1):7, January 2013.

[35] Ritu Garg, Anukriti Bansal, Santanu Chaudhury, and Sumantra Dutta Roy. Text graphic separation in indian newspapers. In *Proceedings of the 4th International Workshop on Multilingual OCR*, MOCR '13, pages 13:1–13:5, New York, NY, USA, 2013. ACM.

[36] Jun-Bao Li, Meng Li, Jeng-Shyang Pan, Shu-Chuan Chu, and John F. Roddick. Gabor-based kernel self-optimization fisher discriminant for optical character segmentation from text-image-mixed document. *Optik - International Journal for Light and Electron Optics*, pages –, 2015.

[37] Hoai Nam Vu, Tuan Anh Tran, In Seop Na, and Soo Hyung Kim. Automatic extraction of text regions from document images by multilevel thresholding and k-

means clustering. In *Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on*, pages 329–334, June 2015.

[38] Pradipta Maji and Shaswati Roy. Rough-fuzzy clustering and multiresolution image analysis for text-graphics segmentation. *Applied Soft Computing*, 30:705 – 721, 2015.

[39] L.A. Fletcher and R. Kasturi. A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:910–918, 1988.

[40] Dov Dori and Liu Wenyin. Vector-based segmentation of text connected to graphics in engineering drawings. In Petra Perner, Patrick Wang, and Azriel Rosenfeld, editors, *Advances in Structural and Syntactical Pattern Recognition*, volume 1121 of *Lecture Notes in Computer Science*, pages 322–331. Springer Berlin / Heidelberg, 1996.

[41] Ruini Cao and Chew Lim Tan. Separation of overlapping text from graphics. In *Proceedings. Sixth International Conference on Document Analysis and Recognition, 2001.*, pages 44 –48, 2001.

[42] Sébastien Adam, Jean-Marc Ogier, and Claude Cariou. Multi-scaled and multi oriented character recognition: an original strategy. In *Fifth International Conference on Document Analysis and Recognition, ICDAR 1999*, pages 45–48. IEEE Computer Society, 1999.

[43] Karl Tombre, Salvatore Tabbone, Loc Plissier, Bart Lamiroy, and Philippe Dosch. Text/graphics separation revisited. In Daniel Lopresti, Jianying Hu, and Ramanujan Kashi, editors, *Document Analysis Systems V*, volume 2423 of *Lecture Notes in Computer Science*, pages 615–620. Springer Berlin / Heidelberg, 2002.

[44] Partha Pratim Roy, Josep Llados, and Umapada Pal. Text/Graphics Separation in Color Maps. *International Conference on Computing: Theory and Applications*, 0:545–551, 2007.

[45] Syed Ali Raza Jafri, Mireille Boutin, and Edward J. Delp. Automatic text area segmentation in natural images. In *ICIP'08*, pages 3196–3199, 2008.

[46] Romain Raveaux, Jean-Christophe Burie, and Jean-Marc Ogier. A colour text/graphics separation based on a graph representation. In *ICPR*, pages 1–4,

2008.

[47] Partha Roy, Umapada Pal, and Josep Lladós. Touching Text Character Localization in Graphical Documents Using SIFT. In Jean-Marc Ogier, Wenyin Liu, and Josep Lladós, editors, *Graphics Recognition. Achievements, Challenges, and Evolution*, volume 6020 of *Lecture Notes in Computer Science*, chapter 18, pages 199–211. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2010.

[48] Thai V. Hoang and Salvatore Tabbone. Text extraction from graphical document images using sparse representation. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 143–150, New York, NY, USA, 2010. ACM.

[49] Sheraz Ahmed, Markus Weber, Marcus Liwicki, and Andreas Dengel. Text / Graphics Segmentation in Architectural Floor Plans. In *11th International Conference on Document Analysis and Recognition.*, 2011.

[50] Thanh-Ha Do, S. Tabbone, and O. Ramos-Terrades. Text/graphic separation using a sparse representation with multi-learned dictionaries. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 689–692, Nov 2012.

[51] C.A.B. Mello and S.C.S. Machado. Text segmentation in vintage floor plans and maps using visual perception. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 3476–3480, Oct 2014.

[52] Nikhil Naikal, Allen Yang, and S. Shankar Sastry. Informative feature selection for object recognition via sparse pca. Technical Report UCB/EECS-2011-27, EECS Department, University of California, Berkeley, Apr 2011.

[53] Duy-Dinh Le and Shinichi Satoh. An efficient feature selection method for object detection. In Sameer Singh, Maneesha Singh, Chid Apte, and Petra Perner, editors, *Pattern Recognition and Data Mining*, volume 3686 of *Lecture Notes in Computer Science*, pages 461–468. Springer Berlin Heidelberg, 2005.

[54] Luka Fürst, Sanja Fidler, and Ale Leonardis. Selecting features for object detection using an adaboost-compatible evaluation function. *Pattern Recognition Letters*, 29(11):1603 – 1612, 2008.

[55] Duy-Nguyen Ta, Wei-Chao Chen, N. Gelfand, and K. Pulli. Surftrac: Efficient tracking and continuous object recognition using local feature descriptors.

IEEE Computer Society Conference on Computer Vision and Pattern Recognition,, 0:2937–2944, 2009.

[56] Peng Fan, Aidong Men, Mengyang Chen, and Bo Yang. Color-surf: A surf descriptor with local kernel color histograms. In *Network Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on*, pages 726–730, Nov 2009.

[57] Jing Fu, Xiaojun Jing, Songlin Sun, Yueming Lu, and Ying Wang. C-surf: Colored speeded up robust features. In Yuyu Yuan, Xu Wu, and Yueming Lu, editors, *Trustworthy Computing and Services*, volume 320 of *Communications in Computer and Information Science*, pages 203–210. Springer Berlin Heidelberg, 2013.

[58] D.G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150–1157 vol.2, 1999.

[59] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.

[60] Faisal Shafait, Daniel Keysers, and Thomas Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. In *Proceedings of the 15th Document Recognition and Retrieval Conference (DRR-2008)*, volume 6815. SPIE, 1 2008.

[61] Sekwon Yeom, Adrian Stern, and Bahram Javidi. Compression of 3d color integral images. *Opt. Express*, 12(8):1632–1642, Apr 2004.

[62] Matthew Brown and David Lowe. Invariant features from interest point groups. In *In British Machine Vision Conference*, pages 656–665, 2002.

[63] StephenM. Smith and J.Michael Brady. Susan a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[64] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[65] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012.

[66] Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *European Conference on Computer Vision (ECCV'10)*, September 2010.

[67] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *ICCV*, volume 2, pages 1508–1515 Vol. 2, Oct 2005.

[68] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *Int. J. Comput. Vision*, 37(2):151–172, June 2000.

[69] Sheraz Ahmed, Marcus Liwicki, and Andreas Dengel. Extraction of text touching graphics using surf. In *Proceedings of 10th IAPR International Workshop on Document Analysis Systems*, DAS '12, pages 349–353, Washington, DC, USA, 2012. IEEE Computer Society.

[70] Rajiv Jain and David Doermann. Logo retrieval in document images. In *2012 10th IAPR International Workshop on Document Analysis Systems*, pages 135–139. IEEE, 2012.

[71] Lu Yang, Gongping Yang, Yilong Yin, and Rongyang Xiao. Sliding window-based region of interest extraction for finger vein images. *Sensors*, 13(3):3799, 2013.

[72] S. Ahmed, F. Shafait, M. Liwicki, and A. Dengel. A generic method for stamp segmentation using part-based features. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 708–712, Aug 2013.

[73] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[74] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[75] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[76] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.

[77] O. Boiman, E. Shechtman, and M. Irani. In defense of Nearest-Neighbor based image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, June 2008.

[78] S. Ahmed, M.I. Malik, M. Liwicki, and A. Dengel. Signature segmentation from document images. In *ICFHR*, pages 425–429, Sept 2012.

[79] Sheraz Ahmed, , Muhammad Imran Malik, Marcus Liwicki, and Andreas Dengel. Extraction of signatures from document images for real world applications. *Journal of the American Society of Questioned Document Examiners*, 18(2):67 – 78, 2015.

[80] M. I. Malik, Marcus Liwicki, Andreas Dengel, and Bryan Found. Man vs. machine: A comparative analysis for forensic signature verification. In *16th Biennial Conference of the International Graphonomics Society (IGS)*, pages 9–13. International Graphonomics Society, 2013.

[81] S. Imade, S. Tatsuta, and T. Wada. Segmentation and classification for mixed text/image documents using neural network. In *Proceedings. 2nd ICDAR*, pages 930 –934, oct 1993.

[82] K. Kuhnke, L. Simoncini, and Zs.M. Kovacs-V. A system for machine-written and hand-written character distinction. In *Proceedings. 3rd ICDAR*, volume 2, pages 811 –814 vol.2, aug 1995.

[83] J.K. Guo and M.Y. Ma. Separating handwritten material from machine printed text using hidden markov models. In *Proceedings. 6th ICDAR*, pages 439 –443, 2001.

[84] Yefeng Zheng, Huiping Li, and D. Doermann. Machine printed text and handwriting identification in noisy document images. *TPAMI*, 26(3):337 –353, march 2004.

[85] Anand Rangarajan, Rama Chellappa, and Anand Rangarajan. Markov random field models in image processing, 1995.

[86] Xujun Peng, Srirangaraj Setlur, Venu Govindaraju, and Ramachandrula Sitaram. Handwritten text separation from annotated machine printed documents using markov random fields. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(1):1–16, 2013.

[87] Sukalpa Chanda, Katrin Franke, and Umapada Pal. Structural handwritten and machine print classification for sparse content and arbitrary oriented document fragments. In *Proceedings of SAC 10*, pages 18–22. ACM, 2010.

[88] Saeed Mozaffari and Parnia Bahar. Farsi/arabic handwritten from machine-printed words discrimination. In *Proceedings. ICFHR*. IEEE, 2012.

[89] Purnendu Banerjee and Bidyut Baran Chaudhuri. A system for hand-written and machine-printed text separation in bangla document images. In *Proceedings ICFHR*. IEEE, 2012.

[90] Sandipan Banerjee. Identification of handwritten text in machine printed document images. In Natarajan Meghanathan, Dhinaharan Nagamalai, and Nabendu Chaki, editors, *Advances in Computing and Information Technology*, volume 177 of *Advances in Intelligent Systems and Computing*, pages 823–831. Springer Berlin Heidelberg, 2013.

[91] Ahmad Montaser Awal, Abdel Belaïd, and Vincent Poulain dAndecy. Handwritten/printed text separation using pseudo-lines for contextual re-labeling. In *14th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 29–34, 2014.

[92] Guangyu Zhu, Yefeng Zheng, David Doermann, and Stefan Jaeger. *Multi-scale Structural Saliency for Signature Detection*, pages 1 – 8. Minneapolis, MN, 2007.

[93] JosL. Esteban, JosF. Vlez, and ngel Snchez. Off-line handwritten signature detection by analysis of evidence accumulation. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(4):359–368, 2012.

[94] Yi Li, Yefeng Zheng, David S. Doermann, and Stefan Jaeger. Script-independent text line segmentation in freestyle handwritten documents. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(8):1313–1329, 2008.

[95] Ranju Mandal, Partha Pratim Roy, and Umapada Pal. Signature segmentation from machine printed documents using conditional random field. In *ICDAR*, pages 1170–1174, 2011.

[96] Guangyu Zhu, Yefeng Zheng, David Doermann, and Stefan Jaeger. Signature Detection and Matching for Document Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2015–2031, November 2009.

[97] Muhammad Imran Malik, Sheraz Ahmed, Marcus Liwicki, and Andreas Dengel. Freak for real time forensic signature verification. In *12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 971–975. IEEE, 2013.

[98] Muhammad Imran Malik, Marcus Liwicki, and Andreas Dengel. Part-based automatic system in comparison to human experts for forensic signature verification. In

12th International Conference on Document Analysis and Recognition (ICDAR), pages 872–876. IEEE, 2013.

[99] Zohaib Khan, Faisal Shafait, and Ajmal Mian. Hyperspectral imaging for ink mismatch detection. In *12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 877–881. IEEE, 2013.

[100] THEO VAN LEEUWEN. What is authenticity? *Discourse Studies*, 3(4):pp. 392–397, 2001.

[101] D.G. Lowe. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, volume 2, pages 1150 –1157 vol.2, 1999.

[102] Sébastien Adam, Jean-Marc Ogier, and Claude Cariou. Multi-scaled and multi oriented character recognition: an original strategy. In *Fifth International Conference on Document Analysis and Recognition, ICDAR 1999*, pages 45–48. IEEE Computer Society, 1999.

[103] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110:346–359, June 2008.

[104] Sébastien Macé, Hervé Locteau, Ernest Valveny, and Salvatore Tabbone. A system to detect rooms in architectural floor plan images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 167–174, New York, NY, USA, 2010. ACM.

[105] Muhammad Imran Malik, Sheraz Ahmed, Faisal Shafait, Ajmal Saeed Mian, Christian Nansen, Andreas Dengel, and Marcus Liwicki. Hyper-spectral analysis for automatic signature extraction. In *17th Biennial Conference of the International Graphonomics Society*, 2015.

[106] Sheraz Ahmed, Muhammad Imran Malik, Faisal Shafait, Ajmal Saeed Mian, Christian Nansen, Andreas Dengel, and Marcus Liwicki. Automatic signature segmentation in
hyperspectral document images. In *Pattern Recognition Letters (submitted)*, 2016.

[107] T. Young. *The Bakerian Lecture: On the Theory of Light and Colours.* Bakerian lecture. Royal Society, 1802.

[108] Robert William Gainer Hunt. *The reproduction of colour*. John Wiley & Sons, 2005.

[109] G Wyszecki and WS Stiles. Color science: Concepts and methods, quantitative data and formulae. 1982. *John Wiley&Sons, New York*.

[110] Chein-I Chang. *Hyperspectral imaging: techniques for spectral detection and classification*, volume 1. Springer, 2003.

[111] D. Lorente, N. Aleixos, J. Gmez-Sanchis, S. Cubero, O.L. Garca-Navarrete, and J. Blasco. Recent advances and applications of hyperspectral imaging for fruit and vegetable quality assessment. *Food and Bioprocess Technology*, 5(4):1121–1142, 2012.

[112] Mihaela Antonina Calin, Sorin Viorel Parasca, Dan Savastru, and Dragos Manea. Hyperspectral imaging in the medical field: Present and future. *Applied Spectroscopy Reviews*, 49(6):435–447, 2014.

[113] Sven Schneider, Richard J. Murphy, and Arman Melkumyan. Evaluating the performance of a new classifier the gp-oad: A comparison with existing methods for classifying rock type and mineralogy from hyperspectral imagery. {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, 98(0):145 – 156, 2014.

[114] Patrick Shiel, Malte Rehbein, and John Keating. The ghost in the manuscript: Hyperspectral text recovery and segmentation. *Codicology and Palaeography in the Digital Age, M. Rehbein and PS und Torsten Schaßan, Eds. Norderstedt: Books on Demand*, pages 159–174, 2009.

[115] Bernard J Aalderink, Marvin E Klein, Roberto Padoan, Gerrit de Bruin, and Tedag Steemers. Clearing the image: A quantitative analysis of historical documents using hyperspectral measurements. In *Poster presented at the AIC 37th Annual Meeting*, 2009.

[116] M. Lettner and R. Sablatnig. Spatial and spectral based segmentation of text in multispectral images of ancient documents. In *10th ICDAR*, pages 813–817, July 2009.

[117] Douglas Goltz, Michael Attas, Gregory Young, Edward Cloutis, and Maria Bedynski. Assessing stains on historical documents using hyperspectral imaging. *Journal of Cultural Heritage*, 11(1):19–26, 2010.

[118] F. Hollaus, M. Gau, and R. Sablatnig. Enhancement of multispectral images of degraded documents by employing spatial information. In *12th ICDAR*, pages 145–149, Aug 2013.

[119] Rachid Hedjam and Mohamed Cheriet. Historical document image restoration using multispectral imaging system. *Pattern Recognition*, 46(8):2297 – 2312, 2013.

[120] F. Hollaus, M. Diem, and R. Sablatnig. Improving ocr accuracy by applying enhancement techniques on multispectral images. In *22nd ICPR*, pages 3080–3085, Aug 2014.

[121] Zohaib Khan, Faisal Shafait, and Ajmal S Mian. Towards automated hyperspectral document image analysis. In *AFHA*, pages 41–45, 2013.

[122] Z. Khan, F. Shafait, and A. Mian. Hyperspectral imaging for ink mismatch detection. In *12th ICDAR*, pages 877–881, Aug 2013.

[123] G Reed, K Savage, D Edwards, and N Nic Daeid. Hyperspectral imaging of gel pen inks: An emerging tool in document analysis. *Science & Justice*, 54(1):71–80, 2014.

[124] Eric B Brauns and R Brian Dyer. Fourier transform hyperspectral visible imaging and the nondestructive analysis of potentially fraudulent documents. *Applied spectroscopy*, 60(8):833–840, 2006.

[125] Aythami Morales, Miguel A Ferrer, Moises Diaz-Cabrera, Cristina Carmona, and Gordon L Thomas. The use of hyperspectral analysis for ink identification in handwritten documents. In *ICCST*, pages 1–5. IEEE, 2014.

[126] Carolina S Silva, Maria Fernanda Pimentel, Ricardo S Honorato, Celio Pasquini, José M Prats-Montalbán, and Alberto Ferrer. Near infrared hyperspectral imaging for forensic analysis of document forgery. *Analyst*, 139(20):5176–5184, 2014.

[127] Sheraz Ahmed, Koichi Kise, Masakazu Iwamura, Marcus Liwicki, and Andreas Dengel. Automatic ground truth generation of camera captured documents using document image retrieval. In *In Proc. of ICDAR*, 2013.

[128] Sheraz Ahmed, Muhammad Imran Malik, Muhammad Zeshan Afzal, Koichi Kise, Masakazu Iwamura, Andreas Dengel, and Marcus Liwicki. A generic method for automatic ground truth generation of camera-captured documents. In *IEEE Transaction on Pattern Recognition and Machine Intelligence (submitted)*, 2016.

[129] Edward Mendelson. ABBYY finereader professional 9.0. *http://www. pcmag. com/article2/0*, 2817(2305597), 2008.

[130] Omnipage ultimate, May 2014.

[131] Ray Smith. An overview of the tesseract OCR engine. In *In Proc. of ICDAR*, volume 2, pages 629–633, 2007.

[132] Thomas M. Breuel. The OCRopus open source OCR system. pages 68150F–68150F–15, 2008.

[133] Alessandro Bissacco, Mark Cummins, Yuval Netzer, and Hartmut Neven. Photoocr: Reading text in uncontrolled conditions. In *ICCV*, pages 785–792, 2013.

[134] Tomohiko Tsuji, Masakazu Iwamura, and Koichi Kise. Generative learning for character recognition of uneven lighting. *In Proc. of KJPR*, pages 105–106, November 2008.

[135] H. Ishida, S. Yanadume, T. Takahashi, I. Ide, Y. Mekada, and H. Murase. Recognition of low-resolution characters by a generative learning method,. *In Proc. of CBDAR*, pages 45–51, 2005.

[136] T. Strecker, J. van Beusekom, S. Albayrak, and T.M. Breuel. Automated ground truth data generation for newspaper document images. In *In Proc. of 10th ICDAR*, pages 1275–1279, July 2009.

[137] Joost van Beusekom, Faisal Shafait, and Thomas M. Breuel. Automated ocr ground truth generation. In *In Proc. of DAS*, pages 111–117, 2008.

[138] T. Kanungo and R.M. Haralick. Automatic generation of character groundtruth for scanned documents: a closed-loop approach. In *In Proc. of the 13th ICPR.,*, volume 3, pages 669–675 vol.3, August 1996.

[139] Tapas Kanungo and Robert M. Haralick. An automatic closed-loop methodology for generating character groundtruth for scanned images. *TPAMI*, 21, 1998.

[140] Gang Zi. GroundTruth Generation and Document Image Degradation. Technical Report LAMP-TR-121,CAR-TR-1008,CS-TR-4699,UMIACS-TR-2005-08, University of Maryland, College Park, May 2005.

[141] Syed Saqib Bukhari, Faisal Shafait, and Thomas Breuel. The IUPR dataset of camera-captured document images. In *In Proc. of CBDAR*, Lecture Notes in Computer Science. Springer, 9 2011.

[142] Jayant Kumar, Peng Ye, and David S. Doermann. A dataset for quality assessment of camera captured document images. In *CBDAR*, pages 113–125, 2013.

[143] Jean-Christophe BURIE, Joseph CHAZALON, Mickael COUSTATY, Sebastien . Eskenazi, Muhammad Muzzamil Luqman, Maroua Mehri, Nibal Nayef, Jean-Marc Ogier, Sophea Prum, and Marcal Rusiol. ICDAR2015 competition on smartphone document capture and ocr (smartdoc). In *Proceedings of 13th ICDAR*, August 2015.

[144] Simon M. Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, Robert Young, Kazuki Ashida, Hiroki Nagai, Masayuki Okamoto, Hiroaki Yamamoto, Hidetoshi Miyao, JunMin Zhu, WuWen Ou, Christian Wolf, Jean-Michel Jolion, Leon Todoran, Marcel Worring, and Xiaofan Lin. ICDAR 2003 robust reading competitions: Entries, results and future directions. *International Journal of Document Analysis and Recognition (IJDAR)*, 7(2-3):105–122, July 2005.

[145] Asif Shahab, Faisal Shafait, and Andreas Dengel. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In *Proc. ICDAR2011*, pages 1491–1496, September 2011.

[146] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere de las Heras. ICDAR 2013 robust reading competition. In *Proc. ICDAR2013*, pages 1115–1124, August 2013.

[147] Dimosthenis Karatzas1, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Lukas Neumann Jiri Matas, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. ICDAR 2015 competition on robust reading. In *Proc. ICDAR2015*, pages 1156–1160, August 2015.

[148] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *In Proc. of ICCVTA*, February 2009.

[149] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning.

In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[150] Robert Nagy, Anders Dicker, and Klaus Meyer-Wegener. NEOCR: A configurable dataset for natural image text recognition. In Masakazu Iwamura and Faisal Shafait, editors, *CBDAR*, volume 7139 of *Lecture Notes in Computer Science*, pages 150–163. Springer Berlin Heidelberg, 2012.

[151] HenryS. Baird. The state of the art of document image degradation modelling. In BidyutB. Chaudhuri, editor, *Digital Document Processing*, Advances in Pattern Recognition, pages 261–279. Springer London, 2007.

[152] Henry S. Baird. The state of the art of document image degradation modeling. In *In Proc. of 4th DAS*, pages 1–16, 2000.

[153] Doe-Wan Kim and Tapas Kanungo. Attributed point matching for automatic groundtruth generation. *IJDAR*, 5:47–66, 2002.

[154] Thomas M. Breuel. A practical, globally optimal algorithm for geometric matching under uncertainty. In *In Proc. of IWCIA*, pages 1–15, 2001.

[155] Joseph Chazalon, Marcal Rusiol, Jean-Marc Ogier, and Josep Llados. A semi-automatic groundtruthing tool for mobile-captured document segmentation. In *Proceedings of 13th ICDAR*, August 2015.

[156] Kazutaka Takeda, Koichi Kise, and Masakazu Iwamura. Memory reduction for real-time document image retrieval with a 20 million pages database. *In Proc. of CBDAR*, pages 59–64, September 2011.

[157] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.

[158] Alex Graves. *Supervised sequence labelling with recurrent neural networks*. PhD thesis, 2008.

[159] Alex Graves, Santiago Fernndez, and Faustino Gomez. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *In Proceedings of the International Conference on Machine Learning, ICML 2006*, pages 369–376, 2006.

[160] Thomas M. Breuel, Adnan Ul-Hasan, Mayce Ibrahim Ali Al Azawi, and Faisal Shafait. High-performance ocr for printed english and fraktur using lstm networks. In *ICDAR*, pages 683–687, 2013.

[161] *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*. IEEE, 2013.

[162] *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*. IEEE, 2013.

[163] ABBYY FineReader, May 2014.

[164] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8, 1966.

[165] S.D. Budiwati, J. Haryatno, and E.M. Dharma. Japanese character (kana) pattern recognition application using neural network. In *In Proc. of ICEEI*, pages 1–6, July 2011.

[166] Ahmed Zaafouri, Mounir Sayadi, and Farhat Fnaiech. Printed arabic character recognition using local energy and structural features. In *In Proc. of CCCA,*, pages 1–5, December 2012.

[167] P. Pavan Kumar, Chakravarthy Bhagvati, and Arun Agarwal. On performance analysis of end-to-end ocr systems of indic scripts. In *In Proc. of DAR*, DAR '12, pages 132–138, New York, NY, USA, 2012. ACM.

[168] S. Sardar and A. Wahab. Optical character recognition system for urdu. In *In Proc. of ICIET*, pages 1–5, June 2010.

[169] Sheraz Ahmed, Markus Weber, Marcus Liwicki, Christoph Langenhan, Andreas Dengel, and Frank Petzold. Automatic analysis and sketch-based retrieval of architectural floor plans. *Pattern Recognition Letters*, 35:91–100, 2014.

[170] Sheraz Ahmed, Marcus Liwicki, Markus Weber, and Andreas Dengel. Improved automatic analysis of architectural floor plans. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 864–869. IEEE, 2011.

[171] A. Heylighen and H. Neuckermans. A case base of case-based design tools for architecture. *Computer-Aided Design*, 33(14):1111 – 1122, 2001.

[172] K. Richter, A. Heylighen, and D. Donath. Looking back to the future - an updated case base of case-based design tools for architecture. *Knowledge Modelling - eCAADe*, 2007.

[173] GH Feng, ZX Sun, and C. Viard-Gaudin. Hand-drawn electric circuit diagram understanding using 2D dynamic programming. *Proceedings of the 11th ICFHR*, pages 493–498, 2008.

[174] Tevfik Metin Sezgin, Thomas Stahovich, and Randall Davis. Sketch based interfaces: early processing for sketch understanding. In *Proceedings of the PUI 2001*, pages 1–8, New York, NY, USA, 2001. ACM.

[175] Levent Burak Kara and Thomas F. Stahovich. Sim-u-sketch: a sketch-based interface for simulink. In *AVI*, pages 354–357, 2004.

[176] Els Den Os and Lou Boves. Towards ambient intelligence: Multimodal computers that understand our intentions. In *eChallenges*, pages 22–24, 2003.

[177] Don Willems, Stephane Rossignol, and Louis Vuurpijl. Mode detection in on-line pen drawing and handwriting recognition. In *Proceedings of the ICDAR '05:*, pages 31–35, Washington, DC, USA, 2005.

[178] Max J. Egenhofer. Spatial-query-by-sketch. *Visual Languages, IEEE Symposium on Visual Languages*, page 60, 1996.

[179] Patrick W. Yaner and Ashok K. Goel. Visual analogy: Viewing analogical retrieval and mapping as constraint satisfaction problems. *Applied Intelligence*, 25(1):91–105, 2006.

[180] S.O. Belkasim, M. Shridhar, and M. Ahmadi. Pattern recognition with moment invariants: A comparative study and new results. *Pattern Recognition*, 24(12):1117 – 1138, 1991.

[181] Bing-Cheng Li and Jun Shen. Fast computation of moment invariants. *Pattern Recognition*, 24(8):807 – 813, 1991.

[182] S. Adam, J.M. Ogier, C. Cariou, R. Mullot, J. Labiche, and J. Gardes. Symbol and character recognition: application to engineering drawings. *International Journal on Document Analysis and Recognition*, 3:89–101, 2000.

[183] Hae Yong Kim and Sidnei Alves de Arajo. Rotation, scale and translation-invariant segmentation-free shape recognition, 2006.

[184] Josep Lladoós, Enric Martí, and Juan Josée Villanueva. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:1137–1143, October 2001.

[185] Luo Yan and Liu Wenyin. Engineering drawings recognition using a case-based approach. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 1*, ICDAR '03, pages 190–, Washington, DC, USA, 2003. IEEE Computer Society.

[186] S. Tabbone, L. Wendling, and K. Tombre. Matching of graphical symbols in line-drawing images using angular signature information. *International Journal on Document Analysis and Recognition*, 6:115–125, 2003. 10.1007/s10032-003-0105-0.

[187] Bruno T. Messmer and Horst Bunke. Automatic learning and recognition of graphical symbols in engineering drawings. In *Selected Papers from the First International Workshop on Graphics Recognition, Methods and Applications*, pages 123–134, London, UK, 1996. Springer-Verlag.

[188] M. Rusiñol and J. Lladós. Symbol spotting in technical drawings using vectorial signatures. In *Graphics Recognition. Ten Years Review and Future Perspectives*, volume 3926 of *Lecture Notes on Computer Science*, pages 35–46. 2006.

[189] M. Rusiñol, J. Lladós, and G. Sánchez. Symbol spotting in vectorized technical drawings through a lookup table of region strings. *Pattern Analysis and Applications*, 13(3):321–331, 2010.

[190] M. Rusiñol, A. Borràs, and J. Lladós. Relational indexing of vectorial primitives for symbol spotting in line-drawing images. *Pattern Recognition Letters*, 31(3):188–201, 2010.

[191] Anjan Dutta, Josep Llados, and Umapada Pal. Symbol spotting in line drawings through graph paths hashing. In *ICDAR-2011 Beijing, China*, 2011.

[192] Nibal Nayef and Thomas M. Breuel. Graphical symbol retrieval using a branch and bound algorithm. In *ICIP*, pages 2153–2156, 2010.

[193] Nibal Nayef and Thomas M. Breuel. Statistical grouping for segmenting symbols parts from line drawings, with application to symbol spotting. In *ICDAR-2011 Beijing, China*, 2011.

[194] Y. Aoki, A. Shio, H. Arai, and K. Odaka. A prototype system for interpreting hand-sketched floor plans. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 747 –751 vol.3, August 1996.

[195] Josep Llads, Jaime Lpez-Krahe, and Enric Mart. A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform. *Machine Vision and Applications*, 10:150–158, 1997. 10.1007/s001380050068.

[196] P. Dosch and G. Masini. Reconstruction of the 3d structure of a building from the 2d drawings of its floors. *Document Analysis and Recognition, International Conference on*, 0:487, 1999.

[197] Philippe Dosch, Karl Tombre, Christian Ah-Soon, and Grald Masini. A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition*, 3:102–116, 2000. 10.1007/PL00010901.

[198] Tong Lu, Huafei Yang, Ruoyu Yang, and Shijie Cai. Automatic analysis and integration of architectural drawings. *International Journal on Document Analysis and Recognition*, 9:31–47, 2007. 10.1007/s10032-006-0029-6.

[199] Siu-Hang Or, Kin hong Wong, Ying kin Yu, and Michael Ming yuan Chang. Abstract highly automatic approach to architectural floorplan image understanding & model generation. 2008.

[200] Raoul Wessel, Ina Blümel, and Reinhard Klein. The room connectivity graph: Shape retrieval in the architectural domain. In V. Skala, editor, *The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2008*. UNION Agency-Science Press, February 2008.

[201] Lluis-Pere de las Heras, Joan Mas, Gemma S?hez, and Ernest Valveny. Wall patch-based segmentation in architectural floorplans. In *ICDAR-2011 Beijing, China*, 2011.

[202] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689 – 694, 1997.

[203] Horst Bunke. Recent developments in graph matching. *ICPR*, 2:2117, 2000.

[204] Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern recognition letters*, pages 255–259, 1998.

[205] BT Messmer and H Bunke. A decision tree approach to graph and subgraph isomorphism detection. *Pattern Recognition*, 32:1979–1998, 1999.

[206] C. Langenhan and F. Petzold. The fingerprint of architecture: Sketch-based design methods for researching building layouts through the semantic fingerprinting of

floor plans. *International electronic scientific-educational journal: Architecture and Modern Information Technologies*, 4 (13), 2010.

[207] Sheraz Ahmed, Markus Weber, Marcus Liwicki, and Andreas Dengel. Text/graphics segmentation in architectural floor plans. In *ICDAR-2011 Beijing, China*, 2011.

[208] Satoshi Suzuki and KeiichiA be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32 – 46, 1985.

[209] Markus Weber, Marcus Liwicki, and Andreas Dengel. Indexing with Well-Founded Total Order for Faster Subgraph Isomorphism Detection. In Xiaoyi Jiang, Miquel Ferrer, and Andrea Torsello, editors, *Graph-Based Representations in Pattern Recognition*, pages 185–194. Springer, 2011.

[210] M. Liwicki, S. El-Neklawy, and A Dengel. Touch & Write - A Multi-Touch Table with Pen-Input. In *DAS*, pages 479–484, 2010.

[211] I.T. Phillips and A.K. Chhabra. Empirical performance evaluation of graphics recognition systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):849 –870, September 1999.

[212] Muhammad Imran Malik, Sheraz Ahmed, Andreas Dengel, and Marcus Liwicki. A signature verification framework for digital pen applications. In *10th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 419–423. IEEE, 2012.

[213] M. Liwicki. Evaluation of novel features and different models for online signature verification in a real-world scenario. In *Proc. 14th Conf. of the Int. Graphonomics Society*, pages 22–25, 2009.

[214] Andreas Schlapbach, Marcus Liwicki, and Horst Bunke. A writer identification system for on-line whiteboard data. *Pattern Recogn.*, 41:2381–2397, July 2008.

[215] A. Piccini, M.and Carpignano and P.C. Cacciabue. Supporting integrated design of control systems and interfaces: A human centred approach. In *Proceedings of the 8th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine-Systems*, pages 87–92, Kassel, Germany, 2001.

[216] Ronan McDonnell Julie Doyle, Zoran Skrba and Ben Arent. Designing a touch screen communication device to support social interaction amongst older adults. In

Proceedings of the 24th BCS International Conference on Human-Computer Interaction (HCI-2010), Dundee, Scotland, 2010.

[217] Holger Koessling, Dominic Gorecky Marcus Liwicki, Markus Weber Gerrit Meixner, and Andreas Dengel. Pen-based interaction forms for smarter product customization. In *Proceedings of the 18th International Federation of Automatic Control World Congress. World Congress of the International Federation of Automatic Control (IFAC-2011),*. IFAC, 4 2011.

[218] L. Michel. Disguised signatures. *Journal of the Forensic Science Society*, 18:25–29, 1978.

[219] Marcus Liwicki, C. Elisa van den Heuvel, Bryan Found, and Muhammad Imran Malik. Forensic signature verification competition 4NSigComp2010 - detection of simulated and disguised signatures. In *12th International Conference on Frontiers in Handwriting Recognition. (ICFHR-2010), November 16-18, India*, pages 715–720, 2010.

[220] Donato Impedovo and Giuseppe Pirlo. Automatic signature verification: The state of the art. *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(5):609–635, September 2008.

[221] Réjean Plamondon and Guy Lorette. Automatic signature verification and writer identification – the state of the art. *Pattern Recognition*, 22:107–131, 1989.

[222] Muhammad Imran Malik, Marcus Liwicki, and Andreas Dengel. Evaluation of local and global features for offline signature verification. In *1st. Int. Workshop on Automated Forensic Handwriting Analysis*, pages 26–30, 2011.

[223] Abdelaali Hassaine and Somaya Al-Maadeed. An online signature verification system for forgery and disguise detection. In Tingwen Huang, Zhigang Zeng, Chuandong Li, and ChiSing Leung, editors, *Neural Information Processing*, volume 7666 of *Lecture Notes in Computer Science*, pages 552–559. Springer Berlin Heidelberg, 2012.

[224] Claudio De Stefano, Angelo Marcelli, and Marco Rendina. Disguising writers identification: an experimental study. In *14th IGS*, pages 99–102, 2009.

[225] Kanghun Jeong and Hyeonjoon Moon. Object detection using fast corner detector based on smartphone platforms. In *Computers, Networks, Systems and Industrial*

Engineering (CNSI), 2011 First ACIS/JNU International Conference on, pages 111 –115, may 2011.

[226] Piotr Bilinski, Francois Bremond, and Mohamed Becha Kaaniche. Multiple object tracking with occlusions using hog descriptors and multi resolution images. In *ICDP*, pages 1–6, 2009.

[227] M. Diem and R. Sablatnig. Recognition of degraded handwritten characters using local features. In *ICDAR*, pages 221–225, 2009.

[228] W. Song, S. Uchida, and Marcus Liwicki. Comparative study of part-based handwritten character recognition methods. In *11th ICDAR*, pages 814–818, 2011.

[229] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, March 1979. minimize inter class variance.

Sheraz Ahmed

PhD Scholar

research highlights

Publications: 17
Citations: 87*
H-index: 7*

research interests

Machine Learning, (Camera-based/Scanned, Hyperspectral) Document Image Analysis, Information Retrieval, Character Recognition, Signature Analysis.

programming

Python, Visual C#, Java, C/C++

* Google Scholar

education

2011–2015 **PhD** Artificial Intelligence — **Kaiserslautern University of Technology, Germany**
A Generic Framework For Information Segmentation in Document Images: A Part-base Approach
Focused on development of generic methods for information (signatures, stamps) segmentation from document images, so that they can be used in verification systems
Completion expected: July 2015

2008–2011 **Masters** of Computer Science — **Kaiserslautern University of Technology, Germany**
Automatic Analysis of Architectural Floor Plans
Developed automatic system, capable of extracting structural and semantic information from scanned floor plan images.

2000–2004 **Bachelors** of Computer Science — **Allama Iqbal Open University, Pakistan**
Real Time Object Tracking & Video Enhancement System
Developed a system for real time object tracking and image enhancement.

fellowships

Sep. 2014 - Nov. 2014 **Visiting Researcher** — **University of Western Australia, Perth, Australia**
Worked on research and development of solutions for automatic segmentation and verification of information from documents using hyperspectral imaging techniques, considering forensic casework and general purpose applications.

Oct. 2012 - Apr. 2013 **Visiting Researcher** — **Osaka Prefecture University, Japan**
Worked on research and development of generic method for automatic ground truth generation of scanned/camera captured document images.

awards and recognitions

Apr. 2011 - till date **PhD Scholarship**
Secured open merit based scholarship for PhD in Kaiserslautern University of Technology (TUKL), Germany, under Promotionsprogram of Department of Computer Science, TUKL

Oct. 2012 - Apr. 2013 **JASSO Scholarship**
Secured Japan Student Services Organization (JASSO) scholarship for research fellowship in Japan

Apr. 2008 **Topped in Masters Computer Science**
First position (nominated for Gold Medal) in the MS(CS) course work at Allama Iqbal Open University , Islamabad , Pakistan

Apr. 2005 **Position holder in Bachelors Computer Science**
Third position in the complete BS (CS) program in the Institute of Computer and Management Sciences, study center of Allama Iqbal Open University.

Apr. 2005 **Best Thesis**
Received best thesis of the session on the final year project of BS(CS) i.e. Real time object tracking and enhancement System

experience

Jul. 2010 - to Date
German Research Center for Artificial Intelligence (DFKI)
Kaiserslautern, Germany
Research Associate
Involved in Research and Development activities in the area of Anoto Technology (Digital pen and Paper Technology). Responsible for design and development of intelligent Forms for Automatic Data extraction for *i*Green Project.

May 2013 - Dec. 2014
b4value.net GmbH
Kaiserslautern, Germany
Software Engineer
Worked as a technology migration expert for different products of b4Value GmbH (a spinoff of DFKI). It includes migration of existing products developed in Java to Visual C#, and development of solutions for information extraction from born digital documents.

May 2009 - Jun. 2011
Fraunhofer Institute for Industrial Mathematics (ITWM)
Kaiserslautern, Germany
Research Associate
Responsible for Development of Graphical User Interface for CoRheoS - Complex Rheology Solver, making Simulation using Visualization Toolkit and Optimization of existing CoRheoS-Grid Algorithm.

Oct. 2008 - Mar. 2009
German Research Center for Artificial Intelligence (DFKI)
Kaiserslautern, Germany
Research Associate
Developed Image Tagging application. Involved in research and development of Automatic Table Recognition from printed documents. Developed preliminary dataset for Table Recognition system.

Sept. 2007 - Aug. 2008
Federal Urdu University of Arts Science and Technology
Islamabad, Pakistan.
Lecturer
Taught different courses of Bachelors of Computer Sciences. Courses includes Discrete Mathematics , Automata Theory , Compiler Construction

Sep. 2006 - Oct. 2007
International Development Research Council (IDRC), Canada
Pakistan Chapter in Islamabad
Research Associate
Conducted Survey on “Accessibility, Acceptance and Effects of Distance Learning Technologies in South Asia” . Performed Analysis of survey results and make recommendations. Worked on development of “Generalized Model for E-assessment system for stu-dents evaluation in South Asia

scientific activities

- Frequent Reviewer for journals; Pattern Recognition, Pattern Recognition Letters, Neural Computing and Applications.
- Frequent Reviewer for refereed int. conferences and workshops including; ICDAR, ICPR, ICFHR, DAS, CBDAR, GCPR, KIS.
- Member Program Committee 4th Int. Workshop on Automatic Forensic Handwriting Analysis 2015, Tunisis.

research & travel grants

Jan. 2014-Dec. 2015 **Hyperspectral Imaging for Automatic Handwriting Analysis and Segmentation** Group of Eight Australia–Germany Joint Research Cooperation Scheme
Active role in finalizing the proposal. Responsible for the hardcore implementation of novel techniques. Funding: **10,000 Euros**.

Jan. 2015-Dec. 2016 **An Intelligent System for Fruits Quality Control** DAAD German–Pakistan Research Collaboration Program
Active role in finalizing the proposal. Part of the team responsible for implementation and completion of the project. Funding available: **80,000 Euros**.

Aug. 2013 **Travel grant from German Research Center for Artificial Intelligence**
12th Int. Conf. on Document Analysis and Recognition (ICDAR), Washington D.C., USA
Funding: **5000 Euros**.

Mar. 2012 **Travel grant from German Research Center for Artificial Intelligence**
10th Int. Workshop on Document Analysis System, Gold Coast, Queensland, Australia
Funding: **3500 Euros**.

Sep. 2011 **Travel grant from German Research Center for Artificial Intelligence**
11th Int. Conf. on Document Analysis and Recognition (ICDAR), Beijing, China
Funding: **3000 Euros**.

references

Prof. Andreas Dengel
Scientific Director
German Research Center for Artificial Intelligence (DFKI), Germany
andreas.dengel@dfki.de

Prof. Marcus Liwicki
Professor
Kaiserslautern University of Technology, Germany
marcus.liwicki@dfki.de

Prof. Faisal Shafait
Assistant Professor
University of Western Australia.
faisal.shafait@uwa.edu.au

publications

[1] Sheraz Ahmed, Markus Weber, Marcus Liwicki, Christoph Langenhan, Andreas Dengel, and Frank Petzold. Automatic analysis and sketch-based retrieval of architectural floor plans. *Pattern Recognition Letters*, 35:91–100, 2014.

[2] Max Feltes, Sheraz Ahmed, Andreas Dengel, and Marcus Liwicki. Improved contour-based corner detection for architectural floor plans. In *Graphics Recognition. Current Trends and Challenges*, pages 191–203. Springer Berlin Heidelberg, 2014.

[3] Muhammad Imran Malik, Sheraz Ahmed, Faisal Shafait, Mian Ajmal Saeed, Andreas Dengel, and Marcus Liwicki. Hyperspectral analysis for automatic signature extraction. In *17th Biennial Conf. of the Int. Graphonomics Society*. 2015.

[4] Sheraz Ahmed, Faisal Shafait, Marcus Liwicki, and Andreas Dengel. A generic method for stamp segmentation using part-based features. In *12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 708–712. IEEE, 2013.

[5] Sheraz Ahmed, Koichi Kise, Masakazu Iwamura, Marcus Liwicki, and Andreas Dengel. Automatic ground truth generation of camera captured documents using document image retrieval. In *12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 528–532. IEEE, 2013.

[6] Sheraz Ahmed, Muhammad Imran Malik, Marcus Liwicki, and Andreas Dengel. Towards signature segmentation & verification in real world applications. In *16th Biennial Conf. of the Int. Graphonomics Society*, pages 139–142, 2013.

[7] Sheraz AHMED, Koichi KISE, Masakazu IWAMURA, Marcus LIWICKI, and Andreas DENGEL. Automatic word ground truth generation for camera captured documents. . *PRMU,* , 112(495):141–146, 2013.

[8] Lluís-Pere de las Heras, Sheraz Ahmed, Marcus Liwicki, Ernest Valveny, and Gemma Sánchez. Statistical segmentation and structural recognition for floor plan interpretation. *International Journal on Document Analysis and Recognition (IJDAR)*, pages 1–17, 2013.

[9] Muhammad Imran Malik, Sheraz Ahmed, Marcus Liwicki, and Andreas Dengel.

FREAK for real time forensic signature verification. In *12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 971–975. IEEE, 2013.

[10] Sheraz Ahmed, Marcus Liwicki, and Andreas Dengel. Extraction of text touching graphics using SURF. In *10th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 349–353. IEEE, 2012.

[11] Muhammad Imran Malik, Sheraz Ahmed, Andreas Dengel, and Marcus Liwicki. A signature verification framework for digital pen applications. In *10th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 419–423. IEEE, 2012.

[12] Sheraz Ahmed, Marcus Liwicki, Markus Weber, and Andreas Dengel. Automatic room detection and room labeling from architectural floor plans. In *10th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 339–343. IEEE, 2012.

[13] Sheraz Ahmed, Muhammad Imran Malik, Marcus Liwicki, and Andreas Dengel. Signature segmentation from document images. In *13th International Conference on Frontiers in Handwriting Recognition*, pages 425–429. IEEE Computer Society, 2012.

[14] Sheraz Ahmed, Marcus Liwicki, Markus Weber, and Andreas Dengel. Improved automatic analysis of architectural floor plans. In *11th International Conference on Document Analysis and Recognition (ICDAR)*, pages 864–869. IEEE, 2011.

[15] Sheraz Ahmed, Markus Weber, Marcus Liwicki, and Andreas Dengel. Text/graphics segmentation in architectural floor plans. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 734–738. IEEE, 2011.

[16] Nazir A. Sangi, Sheraz Ahmed, Sangay Jamtsho, Sonam Rinchen, Sanjaya Mishra, Zeba Khan, V.K. Samaranayake, P. Wimalaratne, K.P. Hewagamage, and Dilhari Attygalle. The emergence of distance education in south asia. In Jon Baggaley and Tian Belawati, editors, *Distance Education Technology in Asia*. SAGE Publications, 2010.

[17] Nazir A. Sangi, Sheraz Ahmed, Sangay Jamtsho, Sonam Rinchen, V.K. Samaranayake, P. Wimalaratne, K.P. Hewagamage, and Dilhari Attygalle. Accessibility, acceptance and effects of distance education in south asia. In Jon Baggaley and Tian Belawati, editors, *Distance Education Technology in Asia*. SAGE Publications, 2010.